Numerical simulation of free surface flows interacting with flexible structures

Zur Erlangung des akademischen Grades Doktor-Ingenieur (Dr.-Ing.)
Genehmigte Dissertation von Jessica Marcela Mariño Salguero aus Riobamba, Ecuador
Tag der Einreichung: 08. April 2021, Tag der Prüfung: 14. Juli 2021

1. Gutachten: Prof. Dr. rer. nat. Michael Schäfer
2. Gutachten: Prof. Dr. rer. nat. Dieter Bothe
Darmstadt

TECHNISCHE
UNIVERSITÄT
DARMSTADT

Mechanical Engineering
Department

FNB

Numerical simulation of free surface flows interacting with flexible structures

Accepted doctoral thesis by Jessica Marcela Mariño Salguero

1. Review: Prof. Dr. rer. nat. Michael Schäfer
2. Review: Prof. Dr. rer. nat. Dieter Bothe

Date of submission: 08. April 2021
Date of thesis defense: 14. Juli 2021

Darmstadt

Cuando es obvio que las metas no pueden ser alcanzadas, no ajustes las metas, ajusta los pasos de acción.

Confucio

To my Ecuadorian family: Pilar, Luis, Elisabeth, and Fercho, and to my German family: Benedikt, Marlies and Heinz for all their love and support.

Erklärungen laut Promotionsordnung

§8 Abs. 1 lit. c PromO

Ich versichere hiermit, dass die elektronische Version meiner Dissertation mit der schriftlichen Version übereinstimmt.

§8 Abs. 1 lit. d PromO

Ich versichere hiermit, dass zu einem vorherigen Zeitpunkt noch keine Promotion versucht wurde. In diesem Fall sind nähere Angaben über Zeitpunkt, Hochschule, Dissertationsthema und Ergebnis dieses Versuchs mitzuteilen.

§9 Abs. 1 PromO

Ich versichere hiermit, dass die vorliegende Dissertation selbstständig und nur unter Verwendung der angegebenen Quellen verfasst wurde.

§9 Abs. 2 PromO

Die Arbeit hat bisher noch nicht zu Prüfungszwecken gedient.

Darmstadt, 08. April 2021

J. Mariño Salguero

Abstract

Efficient numerical methods to simulate free-surface flows interacting with flexible structures are of great interest for enhancing, for instance, the design of marine structures. However, numerical instabilities can occur due to the unsteady free surface flow domain that induces abruptly changing loading conditions on the structure. Common simulation methods also require very small time-steps to accurately capture the free surface flow dynamics, resulting in excessively long computational times. In order to overcome these issues, this work focuses on developing a new efficient and stable free surface flow formulation and its integration in a partitioned fluid-structure interaction (FSI) approach to allow for the simulation of this multiphysics phenomenon.

The free surface flow dynamics are described with the one-fluid formulation of the Navier-Stokes equations and the Volume of Fluid (VoF) method formulated in the arbitrary Lagrangian-Eulerian (ALE) framework. The implementations are made within the in-house finite volume solver FASTEST. First, a more efficient pressure-velocity coupling algorithm is developed. It is an enhanced SIMPLE algorithm with extra correction steps that achieves a better convergence rate of the velocity and pressure fields. Second, the fluid flow interface is captured with an interface capturing scheme implemented through the new Modified Normalized Weighting Factor (MNWF) method. The MNWF method improves the convergence rate and stability of the schemes for medium to high Courant numbers. Thus, larger time-steps can be used, reducing the computational time. Finally, the free-surface flow formulation is integrated into a partitioned FSI approach. The preCICE coupling tool implicitly couples the free surface code of FASTEST with the structural finite element program CalculiX.

The developed solution algorithm and its parts are validated and used to solve benchmark test cases. The obtained results are in perfect accordance with the literature references, and the approach shows a positive effect on accuracy, computational time, and stability.

Zusammenfassung

Effiziente numerische Methoden zur Simulation von freien Oberflächenströmungen, die mit flexiblen Strukturen interagieren, sind beispielweise von großem Interesse für die Verbesserung der Konstruktion von maritimen Strukturen. Allerdings können numerische Instabilitäten aufgrund der instationären Strömungsdomäne an der freien Oberfläche auftreten, die abrupt wechselnde Belastungsbedingungen an der Struktur hervorrufen. Außerdem erfordern gängige Simulationsmethoden sehr kleine Zeitschritte, um die Dynamik der freien Oberflächenströmung genau zu erfassen, was zu übermäßig langen Berechnungszeiten führt. Um diese Probleme zu überwinden, konzentriert sich diese Arbeit auf die Entwicklung einer neuen effizienten und stabilen Berechnungsmethode der freien Oberflächenströmung und deren Integration in einen partitionierten Fluid-Struktur-Interaktion (FSI) Ansatz, um die Simulation dieses Multiphysik-Phänomens zu ermöglichen.

Die Strömungsdynamik der freien Oberfläche wird mit der Ein-Fluid-Formulierung der Navier-Stokes-Gleichungen und der Volume of Fluid (VoF)-Methode beschrieben, die im Arbitrary Lagrangian-Eulerian (ALE)-Rahmen formuliert ist. Die Implementierungen erfolgen innerhalb des hauseigenen Finite-Volumen-Solvers FASTEST. Zunächst wird ein effizienterer Algorithmus zur Druck-Geschwindigkeits-Kopplung entwickelt. Es handelt sich um einen erweiterten SIMPLE-Algorithmus mit zusätzlichen Korrekturschritten, der eine bessere Konvergenzrate der Geschwindigkeits- und Druckfelder erreicht. Danach wird die Strömungsgrenzfläche mit einem Grenzflächenerfassungsschema erfasst, das durch die neue „Modified Normalized Weighting Factor“ (MNWF) Methode, implementiert wird. Die MNWF Methode verbessert die Konvergenzrate und Stabilität der Schemata für mittlere bis hohe Courant-Zahlen. Dadurch können größere Zeitschritte verwendet werden, was die Rechenzeit reduziert. Schließlich wird die Formulierung der freien Oberflächenströmung in einen partitionierten FSI-Ansatz integriert. Das Kopplungstool preCICE koppelt implizit den freien Oberflächen Code von FASTEST mit dem Struktur-Finite-Elemente-Programm CalculiX.

Der entwickelte Lösungsalgorithmus und seine Teile werden validiert und zur Lösung von Benchmark-Testfällen verwendet. Die erhaltenen Ergebnisse stimmen perfekt mit den Referenzwerten aus der Literatur überein und der Ansatz zeigt einen positiven Effekt auf Genauigkeit, Rechenzeit und Stabilität.

Contents

1. Introduction

1.1. Motivation

The fluid-structure interaction (FSI) involving free surface flows plays a dominant effect in the design and operation of many engineering systems, especially in naval architecture, civil engineering, and marine engineering. Marine structures, such as ships and offshore facilities, are massive and capital-intensive structures that can be damaged by the interaction with waves. The overturning, breaking, and collapse of the waves can cause high-frequency vibration of the deck structure, damaging the deck by fatigue. The dropping impact of vast water volumes on the deck induces a relatively large structural deformation as well [90].

The described situation is a multiphysics phenomenon that involves a two-fluid domain with water and air and its interaction with a structure. A domain with two homogeneous, immiscible fluids separated by a sharp interface is a particular case of multiphase flow systems called free surface flows. When free surface flows impact structures and cause deformation, the structural position changes, affecting the surrounding flow dynamics, which again modifies the structural deformation. This phenomenon is called fluid-structure interaction (FSI).

All the aspects of this multiphysics phenomenon have to be considered when designing marine structures. However, this engineering problem is a complex phenomenon with no analytical solution for the most part [107]. Hence, both experimental data and simulations are necessary to enhance the design process and prevent structural damage. Real-scale experiments are costly, and the field measurement data from sensors are not enough to accurately describe the structure damage initiation, and progression [74, 175]. Consequently, numerical simulations are an excellent complement to supply valuable information to describe this phenomenon.

In the last three decades, several numerical methods have been developed that partially solve the problem. Some investigations have focused on creating numerical models that accurately describe the dynamics of free surface water flow around moving rigid bodies [29, 77, 100, 129, 160, 165, 166, 231]. Others have aimed to solve more practical industrial applications. For instance, floating platforms and ships interacting with waves with or without impact phenomena [20, 54, 145, 233], water entry problems in the field of marine hydrodynamics [75, 128], or the interaction of ships with head waves of large amplitude [3, 166, 252]. A good summary of the methods currently applied for ship hydrodynamics is given in [224]. The numerical methods for solving free surface flows in interaction with rigid structures have reached their maturity in development, and nowadays, they are commonly used to solve real industrial applications. For example, in the shipping industry, numerical simulation of flow around a rigid ship moving at a certain speed is used in the design stage to estimate the hydrodynamic resistance of the ship and then minimize it by proper hull-shape design [104, 224]. Researchers are also using these methods to explore propeller propulsion alternatives, such as fin propulsion [161], or to study

marine current turbines [7] and hydraulic turbines [202].

On the other hand, considering the structure as a rigid body does not apply to the design of the new generation of ultra-large container ships. These ships have sizeable open deck areas and thereby highly non-linear wave-induced loading, so hull flexibility plays an essential role in the vessel's response [90]. For these ships, the hull-beam vibration natural frequencies become as low as 0.40 Hz [161]. They can be continually excited due to the high-frequency components in the wave spectrum and non-linear excitation effects. So, to calculate the wave-induced load and prevent fatigue damage to the hull, the flexibility of the structure must be considered.

Similarly, for offshore structures used for sustainable energy production and to extract hydrocarbons and minerals below the seabed at great water depths or in arctic waters, the flexibility of these floating platforms and their support elements is essential to estimate fatigue damages accurately. A good estimation can prevent direct resonance with dominant ocean wave frequencies [161]. Another critical point is the improvement of hydrodynamic calculation procedures involving flow around the slender bodies, which is crucial for advances in offshore wind turbine technology.

A further application for FSI involving free surface flows simulations is investigating the influence of sloshing on the tank structure. Sloshing refers to the dynamic load acting over a tank structure due to a fluid's motion with a free surface confined inside the tank [155]. The numerical results can help to elaborate time-history analysis of storage tanks [249] and thus investigate the influence of the sloshing in the damage of the structural supports as was proposed in [113, 153].

Therefore, the development of consistent, efficient, and stable numerical methods is required to produce higher fidelity outputs for the variables directly linked to the damage of the flexible structures. However, the numerical methods that describe the interaction between free surface flows and elastic structures when the deformation is not neglectable is still a challenge. On the one hand, the structure's continuous deformation introduces rapid changes on the fluid domain grid that induces areas with high local Courant numbers. This condition can destabilize the free surface flow dynamics computation, specifically in air-water flows, requiring very small time-steps to overcome the issue. It results in unpractical long computational times [144]. On the other hand, if the interaction of the free surface flows and an elastic body is strong, the coupling of the two fields describing fluid dynamics and structural dynamics is susceptible to instabilities and decoupling [226]. The abruptly changing load conditions at the fluid-structure interface resulting from the large density differences within the flow domain are among the main reasons for the numerical difficulties [15, 91].

1.2. State of the art

1.2.1. Methods for solving free surface flows

The computation of free surface flows, also called two-fluid flows, is tricky because the topology of the interface is irregular and dynamic. It undergoes abrupt changes, such as melting, tearing, and filamenting, resulting from the physics of two-fluid flows, such as surface tension and phase change [110]. Thus, the fluid dynamics solver requires a particular underlying methodology to model the interface topology. The discussion herein outlines a brief overview of the most significant methodologies developed to compute free surface flows.

In the sixties, Harlow and Welch [79] presented the numerical solution of the so-called dam

breaking problem using a new approach, the **Marker-And-Cell** (MAC) method, opening up the possibility for simulating real free surface flows. In the MAC method, marker particles defined the fluid regions, and the Navier-Stokes equations solved the flow dynamics. The MAC method grew in popularity in the following years and was used to study several problems. For example, the two-fluid Rayleigh-Taylor instability at different density ratios [31], the run-up of a solitary wave on a vertical wall [24], and the transient dynamics of free surface flows in the vicinity of submerged and penetrate obstacles [152]. However, the marker particles sometimes caused inaccuracies, and Hirt and Nichols in 1981 [82], to overcome them, replaced the particles with a marker function. Thus, the **Volume-of-Fluid** (VoF) was born and initiated the next generation of methods for multifluid flows.

The inherent conservation of mass and the applicability to both structure and unstructured grids make the VoF method extremely attractive. However, the discontinuity of the marker function generates numerical diffusion when it advances with the flow. To prevent the marker function from diffusing, the interface is reconstructed so that the marker does not flow into a new cell until the current cell is full. In the initial implementation of VoF, the interface was considered a vertical plane for advection in the horizontal direction and horizontal for the vertical direction. This assumption often led to many small unphysical droplets that break away from the interface, degrading the computation accuracy [203]. In the 1980s, several other, more complex geometrically reconstruction approaches were introduced to improve the interface's representation (see [173] for a comprehensive review of these methods). The most known reconstruction methods are PLIC (for Piecewise Linear Interface Calculation) [130] and SLIC (for Simple Line Interface Calculation) [6]. They accurately approximate the interface shape but are limited for structured grid discretization and require substantial computational effort [245]. Later in the 1990s, Davis [35] introduced a more practical approach, the algebraic reconstruction of the interface that ensures a monotonic change of the volume fraction (boundedness) and reduces the numerical diffusion and dispersion near the interface. Although interface smearing can occur, the implementation is considerably simpler, especially on irregular or three-dimensional grids. The algebraic VoF methods are widely used today due to their robustness in handling complex free surface flow interfaces. For example, in commercial codes for naval hydrodynamics, VoF is included to track overturning, breaking waves, and splashing [224].

The basic idea of the MAC and VoF methods gave rise to other approaches called front-tracking methods. The most known is the **Level-Set** (LS) method introduced by [154], where a continuous level-set function captures the interface movements. The level-set function is defined as a signed normal distance function [188]. The interface is the zero-level set function, and the sign of the function changes across the interface. A marker function with a smooth transition zone from one fluid to another is constructed from the level-set function. It leads to a superior definition of the interface than with the VoF method. However, the accuracy of the method relies on maintaining a constant shape of the level set function near the interface that sometimes is not satisfied. Requiring the application of a reinitialization procedure [198] to overcome the problem. The level-set function is always adjusted so that its value equals the shortest distance to the interface. This reinitialization is commonly believed to be the primary source of the weak mass conservation [7].

The sharp interface definition and the ability to model changes in free surface topology have led to improving the method's mass conservation. Some technique is combining LS with VoF

considering the VoF function when the interface is advected [141, 197, 215] or coupling the standard LS and the conservative LS method [250]. A detailed list of more techniques can be found in [188]. In water-air problems, the most used level set method is the single-phase approach [81]. It only computes the water region and extrapolates the air region applying the free surface boundary condition, resulting in less computational time.

Another alternative to solve free surface problems is to apply the fitting methods in which the grid is continuously regenerated to fit each current free surface shape. A technic in this group is the **Arbitrary Lagrangian-Eulerian** (ALE) method. The free surface is approximated as a mesh boundary that moves influenced by the flow. The pressure of the air is assumed constant, and the viscous stresses in the air are negligible. Then, the jump conditions (no phase-change, no-slip at the interface, and force balance) become a boundary condition for the liquid domain [203]. An advantage of the fitting methods is that the form and location of the free surface are explicitly known, and it is always sharp. Thus, more details of the flow near the wave surface are known, and the accuracy of force computation increases [84]. The ALE method is very suitable for simulating free surface problems with large deformation of the interface, but without topological changes [205, 228]. Computing breaking waves, overturning, and splashing phenomena become a challenging task for the ALE method due to the considerable topological changes of the interface that lead to a degeneration of the computational mesh [242]. The remeshing techniques are an additional source of errors since quantities of interest have to transfer from the old mesh to the new mesh [10]. For this reason, in industrial applications, it is usually applied for the prediction of steady flow around ship hulls or flows with mild breaking only [224].

A further strategy introduced in recent years is the use of particles to track the movement of the interfaces. A set of independent particles discretizes the computational domain, and the grid that the particles form can adapt to account for any changes in the interfacial shape [195, 214]. The most representative methods in this category are the **Smoothed Particle Hydrodynamics** (SPH) method initially developed for astrophysical computations [69], and the **Moving Particle Semi-implicit** (MPS) [109] method. In the SPH method, pure Lagrangian large particles subject to Newton's Second Law represent the fluid, and analytical differentiation of the interpolation functions estimate the spatial derivatives. Therefore, a fixed computation grid is not necessary, and the SPH particle positions and attributes are calculated using the standard numerical integration methods in the time domain [75]. The MPS method is similar to the SPH method. However, MPS applies simplified differential operators models, and the unsteady solution of the governing equations is obtained through a semi-implicit prediction-correction process. The implementation of these meshless methods is straightforward and mass conservative. Nevertheless, the accuracy is limited for multidimensional flows and depends on the number of particles used. Another disadvantage of the method is the difficulty in correctly modeling the boundaries.

1.2.2. Fluid-structure interaction numerical approaches

Fluid-structure interaction refers to the physical phenomenon produced when a flow induces compression and shear forces on the structure's surface that moves or deforms the structure. The structural position changes affect the surrounding flow dynamics, which changes the compression and shear forces on the structure. It results in a continuous process where each flow adaptation also induces an adaptation of the structure position [150]. Examples of fluid-structure interactions are parachutes [190, 199], airbags [103], blades of a turbo-machine [72, 76], heart valves [163,

218], floating structures interacting with waves [20, 247]. The physical coupling of the fluid field and structural field is mathematically modeled as a dynamic equilibrium at the fluid-structure interface.

There are two main numeric approaches to simulate FSI problems, the monolithic approach and the partitioned approach.

The monolithic approach solves the fluid, structure, and interface equations simultaneously in a single and specialized solver (see, e.g., [12, 14, 80, 90, 91, 235]). The procedure results in an inherent coupling between the subfields. Thus, mapping and information exchanges between the subfields computational domains are not required.

In contrast, in the partitioned approach, the domain is decomposed in non-overlapping domains for the flow and the structure. Two separate field solvers solve the fluid dynamics and the structure dynamic in an iterative process and explicitly use the interfacial conditions to relate information between fluid and structure solutions (see, e.g., [26, 42, 101, 186, 220]). The Lagrangian formulation is often used for the solid part since the deformation has to be determined from a known reference configuration, which can be done by tracking the corresponding material points. While the Arbitrary Lagrangian-Eulerian (ALE) formulation is usually applied for the description of the fluid dynamics, it allows the Eulerian description of the pressure and velocity fields in moving domains [155, 185]. Also, generally, the mesh for the flow domain is much finer than for the structure domain. The main challenge with this approach is the coupling of the solvers at the fluid-structure interface, which requires technics to exchange information over the interface and the corresponding interpolation algorithms to interpolate the loads and displacements between the non-matching meshes [36, 87].

The monolithic methods usually are more robust and sometimes more efficient than partitioned schemes [39, 236]. However, the formulation must be adapted to each problem, so it is less versatile than the partitioned approach. Thus, nowadays, the partitioned approach is the most applied approach due to its modularity and the possibility of re-using existing software (see, e.g., [2, 52, 55, 71, 101, 139]). Since their flexibility, also further physical effects can easily be added to the FSI application. Such as heat transfer in the analysis of thermal fatigue [97], acoustics effects for noise predictions [50, 127], or free surface flows for the damage prediction of partially filled LNG tanks [113].

1.2.3. The partitioned fluid-structure interaction approach

In partitioned fluid-structure interaction problems, the solvers act as black-boxes with a given input-output relation at the common boundary. Regularly, the fluid solver receives velocities as an input, solves the flow system, and returns forces as an output. The structure solver receives forces as an input, computes the structural deformations, and returns displacements [19]. The coupling conditions at the coupling interface are the kinematic condition (the balance of displacements and velocities) and the dynamic condition (the balance of forces). The accomplishment of them prevents the inherent added-mass effect of the partitioned approaches [38].

The coupling conditions are not part of the large system of the fluid or structural equations. They form a highly non-linear system of equations called fixed-point equations, which can be solved with an explicit scheme or with an implicit scheme depending on the strength of the physical coupling between the involved single-physics fields [140]. Explicit schemes, sometimes called ”weak couplings,” only perform a fixed amount of solver calls during one time-step. In

contrast, implicit schemes or ”strong couplings” iteratively exchange data between the solvers until the fully coupled solution is recovered. The explicit schemes are faster than implicit schemes, but they are restricted to small time-steps [38] and often lack accuracy and numerical stability because the energy is not exactly balanced [42].

On the contrary, the implicit schemes are numerically stable but are compute-intensive schemes. A traditional solution of the fixed-point system is the staggered serial scheme based on the Gauss-Seidel execution of both solvers. This scheme has poor convergence and is computationally expensive because it allows a fully uncoupled (sequential) solution of the discrete problem [42, 56, 140]. So, it is often applied together with Aitken's dynamic relaxation to accelerate convergence [101, 111, 227]. Another option to faster solve the fixed-point system is the Newton-based methods (see, e.g., [39, 126]). In the Newton methods, the solution of the linearized fixed-point equations only requires the repeated residual evaluation of the interface displacements; in other words, the repeated evaluation of the Jacobian of the coupled operator. Due to usually the Jacobian is approximated rather than exact evaluated to reduce computational time, these methods are so-called inexact (or quasi-) Newton methods (see, e.g., [39, 56–58]). The quasi-Newton methods allow parallel execution of the solvers, yielding more efficient and robust coupling schemes (see, e.g., [18, 140, 212]).

1.2.4. Strategies to deal with the dynamic fluid-structure interface

In an FSI computation, the fluid domain needs to follow the structure deformation, requiring moving boundaries methods (These are very similar to the methods used to track the free surface flow interface). Usually, the moving domain is addressed with fitting methods as the Arbitrary Lagrangian-Eulerian (ALE) method [3, 101], or with Eulerian methods, also called fixed grid methods. In the ALE method, the boundary grid points must follow the fluid-structure interface movements (thus the Lagrangian terminology). In contrast, the remaining points move not necessarily concerning the boundary kinematics (thus the Eulerian terminology) [56, 101]. The fitting methods accurately approximate wall shear stress on the fluid-structure interface due to the fluid and solid domains do not overlap [38]. However, mesh movement strategies are required, and they can often not support substantial deformation. Some strategies are the spring analogy, the elastic medium analogic, or re-meshing. A summary of them is given in [9, 241].

On the contrary, the Eulerian methods, also called fixed grid methods, support highly deformed structures. The immersed boundary (IB) method of Peskin [164] and its variants (see, e.g. [90, 125, 223, 251]) are representant of them. Peskin initially developed the IB method to study the fluid dynamics of heart valves [163]. The IB discretizes the fluid domain on a fixed Cartesian mesh, and the Lagrange solid nodes move freely through the fixed fluid mesh without being forced to adapt to it in any way. Near the solid nodes, a weighting function that involves a smoothed approximation of a Dirac delta function interpolates the elastic forces from the Lagrangian solid grid to the Eulerian fluid grid. This interpolation introduces an additional body-force source term in the momentum equation. Another fixed grid method is the Fictitious Domain method developed in [70, 157]. It has been used for the simulations of cardiac valves [218, 219] and the study of the iteration between two fluids and moving rigid bodies [160]. In this method, the structure is considered a fictitious fluid in the fluid domain. Hence, the Navier-Stokes equations are solved in the entered computational domain, and then to conserve linear and angular momentum, the rigid body velocity is imposed in the solid region. The fixed grid methods can deal with arbitrarily

large movements and topology changes, but they lost accuracy near the interface caused by the interpolations [38].

1.2.5. Aproaches for solving free surface flows interacting with flexible structures

Free surface flows interacting with flexible structures is a multiphysics problem with high non-linearities that requires robust computational methods. Interesting monolithic and partitioned FSI approaches involving free surface flows have been developed in recent decades. In the following, some of these approaches are presented.

The two pioneering monolithic approaches were the FEM-based approach combined with the level set method presented by Walhorn et al. [226] in 2005, and the Lagrangian FSI approach where the solid and the two-fluid flow were described by the SPH method of Antoci et al. [5] in 2007. Walhorn et al. introduced the breaking dam on the elastic wall problem. Antoci et al. presented the numerical and experimental solution of the deformation of an elastic plate subjected to time-dependent water pressure. Nowadays, the two cases are considered benchmark cases and are used to validate new approaches.

In the following years, many other monolithic approaches were developed, mainly using Lagrangian particle methods due to their potential to model extreme deformation of materials [121]. For instance, an approach based on particle finite element method (PFEM) [91], an SPH projection method [168], an SPH-FEM approach [237], a coupled MPS-modal superposition method [195], a modified MPS method with the large eddy simulation (LES) method [234], an enhanced MPS with stress point integration [48], a coupling of SPH with smoothed point interpolation method [248], and a coupled SPH with the volume compensated particle method (VCPM) [151].

On the other hand, several partitioned FSI approaches have also been developed. Kassiotis et al. [101] coupled the finite element program FEAP and the finite volume program OpenFOAM through the Communication Template Library (CTL). FEAP solves the structure deformation, and OpenFOAM solves the free surface flow using a VoF method formulated in an ALE framework. Kangping and Changhong [122] coupled a FEM solver for the structural deformation and a finite difference (FD) solver for the fluid domain. An FD method based on regular fixed Cartesian grids and an improved tangent of the hyperbola method [232] captures the free surface motion. The immersed boundary method is adopted to couple the FDM and the FEM. Also, Paik and Carrica [155] proposed an FD-FEM partitioned approach. However, a difference of [122], the gluing method associates the non-matching grids between fluid and structure domains. Further, the free surface is modeled with a single-phase level set method. Equally, Yang et al. [234] developed a 2D implicit approach. The formulation relies on an immersed structural potential method, including a level set-based method for capturing the fluid-flow interface and an interface Lagrangian-based meshless technique for tracking the fluid-structure interface.

Although several algorithms that can address free surface flows with elastic structures have been developed, there are still many challenges to combining the two fields and simulating more complex real systems in a reasonable time. This is mainly due to the limitations of solving algorithms in getting stable and achieving high accuracy when strong interactions between the fields exist. For example, in [123], the dam break's simulation results impacting an elastic plate were contrasted to the experimental results. The numerical solution captured well the main

features of the phenomena at the initial impacting stage. However, more accuracy in predicting is necessary for the higher modes of vibration.

Similarly, in the usual validation test case used by the based Lagrangian particle methods, the rolling tank with an elastic bar interacting with a free surface, the bar deformation and free surface elevation results typically agree with the experiments. However, the bar deformations exhibited a greater amplitude than the experiments, possibly caused by the three-dimensional effects overlooked in these approaches. For more practical applications as simulations of floating offshore wind turbines, the numerical results still have significant differences in damage patterns compared with the laboratory fatigue test [13].

In summary, the literature review reveals that there is still a necessity to investigate and develop more stable, efficient, and accurate approaches for solving free surface flows in interaction with elastic structures. It entails better and more flexible fluid solvers, structural solvers, and multiphysics coupling tools that quickly reproduce accurate results.

1.3. Scope and Objectives

A reliable numerical solution for solving free surface flows in interaction with elastic structures requires a consistent solution of each part of the multiphysics phenomenon. In the fluid part, velocity and pressure fields must be adequately solved and moving boundaries considered. Besides, the formulation needs to include a method to capture the free surface interface. In the structural part, the computation of stresses and strains requires consistent elastic models to address large deformations. Furthermore, a consistent FSI approach needs to be applied.

According to the literature review, many structural methodologies can correctly model the elastic structural part of the problem. In contrast, there is still the necessity to investigate and develop more efficient and accurate free surface flows solvers that remain stable, even using large time-steps. Additionally, FSI coupling approaches are needed that can manage high non-linearities without lacking stability.

Therefore, this thesis focuses on developing an efficient and stable two-fluid flow solver and its integration into a partitioned FSI approach. The partitioned approach is chosen due to its versatility and the possibility to use specialized field solvers. The two-fluid flow formulation will be implemented in the in-house finite volume solver FASTEST. Meanwhile, the elastic structural part will be solved with the open-source code CalculiX, based on the finite element method (FEM).

This research is divided into three targets related to improving and upgrading the FASTEST solver to efficiently solve two-fluid flow problems and use it in an FSI approach.

The first target is improving the pressure-velocity coupling algorithm since this is the most time-consuming part of the computation. A more efficient algorithm that achieves better convergence rates of the velocity and pressure fields can drastically accelerate the simulations.

The second central target is developing and implementing a stable and efficient two-fluid flow solver based on an algebraic VoF method to capture sharp free surface interfaces even at high Courant numbers conditions to be able to use large time-steps and reduce the computational time.

The third target is designing an FSI algorithm that couples the new two-fluid flow solver developed in FASTEST with the structural solver CalculiX via the multiphysics coupling tool preCICE. The ALE technique addresses the moving boundaries in the fluid domain.

Finally, different benchmark test cases from the literature will be solved to verify and validate each new implementation and the complete approach.

1.4. Outline

The thesis is structured as follows. Chapter 2 contains the mathematical modeling of the multi-physical phenomenon. The chapter starts with a general overview of the problem and the different mathematical descriptions of motion. The Eulerian, Lagrangian, and Arbitrary Lagrangian-Eulerian approaches are presented. The chapter reviews the formulation of two-fluid flow's governing equations, followed by describing the structural dynamics. The chapter closes with the presentation of the FSI coupling conditions.

As the algorithm used in the CalculiX code is out of the research scope, the numerical discretization process based on the FEM is not detailed here. For more information about this topic, refer, for example, to [11]. On the other hand, the implementations in the in-house FASTEST code will be shown in detail throughout the text.

In Chapter 3, the numerical solution of the two-fluid flow governing equations is discussed. The principal focus is the finite volume discretization methodology of the Navier-Stokes equations in the FASTEST code. Special attention is given to the pressure-velocity coupling algorithm. At the end of the chapter, a new, more efficient pressure-velocity coupling algorithm is presented with its verification and validation procedure.

Chapter 4 deals with the discretization of the VoF transport equation. First, some background about high-resolution advection schemes and interface capturing schemes are outlined. Second, the chapter introduces the new methodology adopted for the numerical implementation of the interface capturing schemes. Third, the complete two-fluid flow solver is described and validated, solving different benchmark cases. The chapter ends with a performance analysis of three interface capturing schemes suitable for use in FSI applications.

The two-fluid flow solver developed in Chapters 3 and 4 is coupled to the partitioned FSI approach in Chapter 5. The chapter begins with some relevant background of the FSI partitioned approach and its redesign. It includes information about the communication and data mapping, the implicit coupling scheme, the acceleration techniques, and coupling configuration. Then, a description of the FASTEST adapter developed in this thesis's scope is provided, as well as a validation FSI test. The chapter finalizes presenting the solution of three cases of free surface flows interacting with flexible structures.

Finally, Chapter 6 concludes the thesis results and presents an outlook on future research.

2. Mathematical modeling of the multiphysical phenomenon

This chapter focuses on the mathematical modeling of the interaction between an elastic structure and free surface flows. This multiphysics transient phenomenon can be exemplified by the interaction of a ship and waves in the sea, as illustrated in Figure 2.1. The domain Ω involves a two-fluid flow subdomain and a structural subdomain denoted by $\Omega^{\mathrm{f}}_{1,2}$ and Ω^{s}, respectively. A well-defined interface represented by Γ^{fi} separates the two fluids, which are generally immiscible and distinct in their composition. The fluid-structure interface Γ^{fsi} is the boundary where the flexible structure interacts with the two-fluid flow. The principle of conservation of mass and momentum governs the dynamics of the two-fluid flow domain, while the structure deformation follows the principles of an elasticity model. The two subdomains of the problem are separately addressed with different approaches and then coupled at the fluid-structure interface. The Fluid subdomain is modeled with the Arbitrary Lagrangian-Eulerian (ALE) approach and the structural subdomain with the Lagrangian one.

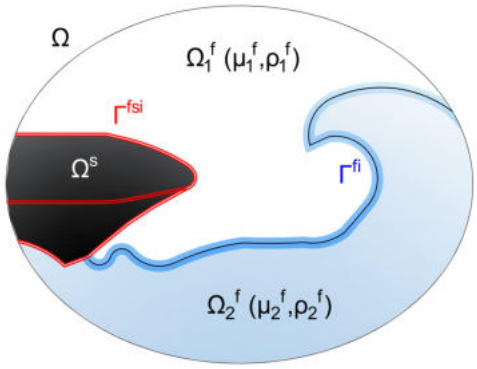


Figure 2.1.: Problem domain of the interaction between a ship and waves, Ω^{s} refers to the structure and $\Omega^{\mathrm{f}}_{1,2}$ the two-fluids. ρ is the density and μ the viscosity of each fluid. The blue line is the interface between the two-fluids, and the red line the fluid-structure interface

The chapter begins with an overview of the different approaches for developing the mathematical formulation of the conservation laws in Section 2.1. The discussion of the approaches is addressed from a numerical point of view. Afterward, Section 2.2 presents the mathematical model for the two-fluid flow subdomain formulated using the ALE approach. Section 2.3 clarifies the development of the elastic model in the Lagrangian approach. The definition of the fluid-structure

interface condition and general boundary conditions close the chapter in Section 2.4.

Note that the **BOLD** uppercase characters symbolize tensors or matrixes, and **bold** lowercase letters symbolize vectors throughout the document.

2.1. Lagrangian, Eulerian, and ALE approaches

In continuum mechanics, the Lagrangian, Eulerian, and ALE approaches describe the changes of material properties during the motion and express the conservation laws mathematically. Usually, the Lagrangian approach is mainly employed in structural mechanics. In contrast, the Eulerian approach is prevalent to solve fluid mechanics, and the ALE description applies for formulating significant distortion responses of materials and fluid-structure interaction problems.

In the Lagrangian or material description, the continuum is subdivided into material points linked to grid points, as is schematized in Figure 2.2a. Each point of the grid follows the movement of the associated material point along its pathline [185]. The properties are a function of the material point position in time, referred upon its initial configuration, which facilitates the treatment of materials with history-dependent behavior. So, it is preferred for solid mechanics. However, since the computational mesh moves with the material, when significant distortions in the motion occur, e.g., vortices in fluids, frequent remeshing is necessary, causing a loss of accuracy [45].

On the other hand, the Eulerian approach or spatial description divides the domain into regions and measures in each specific region the changes of properties of the material particles crossing it [147]. The material properties have only instantaneous significance in a fixed region of space and are expressed in terms of position and time without any reference to the initial configuration. The computational mesh is fixed, and the grid nodes are dissociated from the material particles (see Figure 2.2b). It facilitates the handling of complex material movements such as turbulent flows. Nevertheless, this description includes convection effects in the formulation, which increases the numerical difficulties.

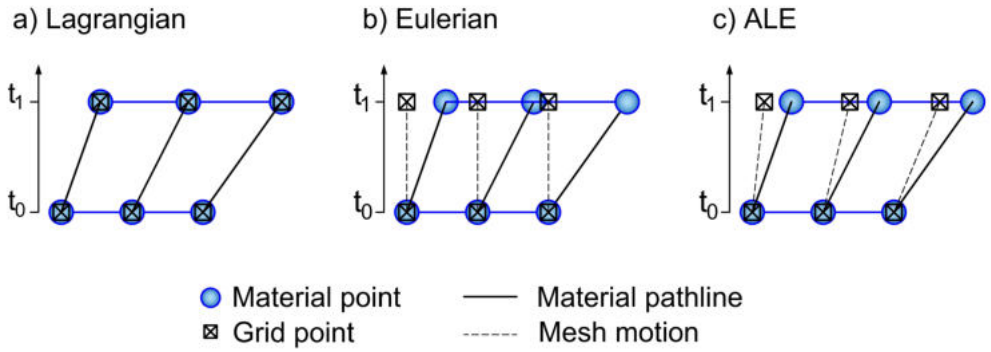


Figure 2.2.: Advance in time of three material points and computational grid points according to the a) Lagrangian, b) Eulerian, and c) ALE description

The Arbitrary Lagrangian-Eulerian approach is a generalization of the Lagrangian and the Eulerian approaches and attempts to overcome their weaknesses. The ALE description considers

a third domain called the ALE map as a reference domain, mapped from either the material domain (Lagrangian description) or the spatial domain (Eulerian description). This third domain describes the evolution of the grid during the domain movements and is used to write the partial time derivative in terms of a reference fixed configuration. In other words, the ALE map associates, at each time, a point in the current computation domain to a point in the reference domain [62]. The ALE domain moves somehow arbitrary, as is shown in Figure 2.2c with a convective velocity defined as

$$\mathbf{c} = \mathbf{v} - \mathbf{v}_g, \tag{2.1}$$

where $\mathbf{c}$ is the relative velocity between the material velocity $\mathbf{v}$ and the mesh velocity $\mathbf{v}_g$. The freedom in moving the computational domain in the ALE approach allows a more significant deformation of the material than the Lagrangian approach [165] with more precision than the Eulerian approach. A comprehensive explanation of the ALE approach is given in reference [45]

2.2. Two-fluid flow domain

The fluid domain $\Omega^{\mathrm{f}}_{1,2}$ is a multifluid flow formed by two isothermal, viscous, immiscible, and incompressible Newtonian fluids. However, for modeling purposes, the two fluids are treated as a single continuum fluid with varying physical properties that change abruptly across the fluid interface. This simplification of the two-fluid modeling is known as the one-fluid formulation of the Navier-Stokes equations. The one fluid formulation describes the multifluid dynamics introducing the stress concentrated at the sharp interface, called surface tension, as an additional body force in the momentum equation and tracking the instantaneous position of two fluids with an additional indicator function [203].

Moreover, because the unsteady fluid-structure interface is a moving boundary in the fluid domain, the governing equations are formulated in the ALE framework. So, the mesh velocity $\mathbf{v}_g$ is included in the model equations. The space conservation law (SCL) introduced in [201] completes the fundamental principles of the fluid domain motion. The SCL prevents the appearance of artificial mass sources due to the domain movement [41, 131, 138], thus maintaining the global conservation of the system.

2.2.1. Mass Conservation (The continuity equation)

The mass conservation principle states that the mass flow is neither created nor destroyed. The differential form of the mass conservation or continuity equation in the ALE framework is given as

$$\left.\frac{\partial \rho}{\partial t}\right|_{\mathrm{ALE}} + \nabla \cdot \left[\left(\mathbf{v} - \mathbf{v}_g\right)\rho\right] = 0, \tag{2.2}$$

where ρ is the density, t denotes time, $\mathbf{v}$ is the fluid velocity vector, and $\mathbf{v}_g$ is the velocity with which the grid moves. For a fixed domain, $\mathbf{v}_g = 0$, and the formulation adopts the Eulerian description.

For incompressible flows and fulfilling the space conservation law, which will be addressed later, the continuity equation reduces to

$$\nabla \cdot \mathbf{v} = 0. \tag{2.3}$$

2.2.2. Momentum Conservation

The principle of conservation of momentum implies that the action of external force changes the momentum of a body. Its governing equation is Newton's second law of motion, which, in the ALE framework, has the derivative form

$$\rho \left.\frac{\partial \mathbf{v}}{\partial t}\right|_{\text{ALE}} + \rho \nabla \cdot \left[\left(\mathbf{v} - \mathbf{v}_{\text{g}} \right) \mathbf{v} \right] = \nabla \cdot \tau + \mathbf{b}. \tag{2.4}$$

The tensor τ represents the stress, and vector $\mathbf{b}$ the total body forces applied to the fluid.

For Newtonian incompressible fluids, the stress tensor is assumed to be a linear function of the rate of strain tensor [60] and is defined with the Stokes law as

$$\tau = -p\mathbf{I} + \mu \left\{ \nabla \mathbf{v} + (\nabla \mathbf{v})^T \right\}, \tag{2.5}$$

where μ is the dynamic viscosity, p is the pressure, and $\mathbf{I}$ is the identity tensor.

Moreover, the body force term for multiphase flows is the sum of the gravitational forces and the surface tension effects given by

$$\mathbf{b} = \rho \mathbf{g} + \sigma \kappa \mathbf{n}^{\text{fi}} \left| \nabla \alpha \right|. \tag{2.6}$$

The vector $\mathbf{g}$ is the gravity, the constant σ is the surface tension coefficient, and κ is the interfacial curvature (twice the mean interface curvature). The vector $\mathbf{n}^{\text{fi}}$ is the outward normal to the interface between the two fluids Γ^{fi}, and the indicator scalar function α is the volume fraction of one of the fluids. The surface tension is expressed as a volume force acting in a normal direction towards the fluid interface following the continuum surface method (CSF) introduced by Brackbill, Kothe, and Zemach [17].

The influence of the gravitational forces or the surface tension on the flow dynamics depends on the flow interface geometry. When the radius of the interface curvature tends to infinity, e.g., water waves, the curvature tends to zero, and the surface tension vanishes. On the contrary, for small radii, e.g., a small water droplet, κ becomes very significant, and thus the surface tension is dominant, and the gravity force is insignificant [203].

Substituting the expressions of the stress tensor Eq. (2.5) and body forces Eq. (2.6) into the momentum equation (2.4) and assuming constant viscosity results in

$$\rho \left.\frac{\partial \mathbf{v}}{\partial t}\right|_{\text{ALE}} + \rho \nabla \cdot \left\{ \left(\mathbf{v} - \mathbf{v}_{g} \right) \mathbf{v} \right\} = -\nabla p + \nabla \cdot \mu \left[\nabla \mathbf{v} + (\nabla \mathbf{v})^T \right] + \rho \mathbf{g} + \sigma \kappa \mathbf{n}^{\text{fi}} \left| \nabla \alpha \right|. \tag{2.7}$$

This expression represents the equation of motion in the convective or non-conservative form. It is mainly used for incompressible flows with large density jumps because the conservative form can lead to numerical difficulties. These troubles are mathematically demonstrated in [203].

Additionally, notice that the momentum equation (2.7) uses average values of the material properties that are recovered from the volume fraction α and the individual material properties of each constituent fluid by

$$\rho = \alpha \rho_1 + (1 - \alpha) \rho_2, \;\; \mu = \alpha \mu_1 + (1 - \alpha) \mu_2, \tag{2.8}$$

which are algebraic statements of local mass conservation [110].

2.2.3. The indicator function (Volume of Fluid method)

The volume of fluid (VoF) method developed by Hirt and Nicholas [82] is a well-known conservative method designed to capture the unsteady evolution of the flow interface between two fluids. In the VoF method, the volume fraction of one of the fluids denoted by α defines the spatial distribution of the fluids. α is a discontinuous scalar indicator function that takes values from zero to one. $\alpha = 0$ denotes absence, $\alpha = 1$ denotes the presence of the traced fluid, and $0 < \alpha < 1$ indicates a mixture. The value of $\alpha = 0.5$ defines the flow interface.

The VoF method introduces an additional transport equation to advect the volume fraction, which in the ALE framework has the form

$$\left.\frac{\partial \alpha}{\partial t}\right|_{\text{ALE}} + \nabla \cdot \left[\left(\mathbf{v} - \mathbf{v}_g\right) \alpha\right] = 0. \tag{2.9}$$

2.2.4. The space conservation law

The SCL results from the integral statement of the mass conservation law for a spatial region of volume V limited by an enclosed surface S and is written as

$$\left.\frac{\partial}{\partial t}\right|_{ALE} \int\limits_{V(t)} dV - \int\limits_{S(t)} \mathbf{v}_g \cdot \mathbf{n} dS = 0. \tag{2.10}$$

The time integration schemes have to fulfill the space conservation law (SCL) to prevent the added-mass effect [65] and preserve the non-linear stability properties of the temporal discretization schemes [41, 53, 68].

2.3. Elastic structural domain

When the structure's elastic deformation is a partial aspect of the fluid-structure interaction coupling, an elasticity model must be applied to model the structural domain mathematically. This model describes how the solid object reversibly deforms and becomes internally stressed due to flow-induced loads.

An elasticity model builds on the *kinematics, Newton's second law, and a constitutive law*. The model determines all the stresses, strains, and displacements that the elastic body experiences after deforming from its reference state. A comprehensive explanation of the elastic model can be found in [44, 64].

2.3.1. Kinematics

The kinematics describes the motion of the object and represents the relationship between displacement and strain. The motion can be a translation, rotation, or deformation of the solid body. The Lagrangian specification of the solid motion of a material particle P is illustrated in Figure 2.3.

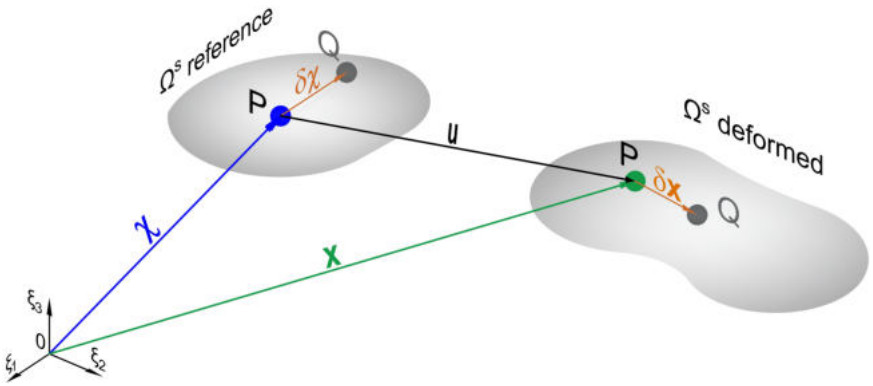


Figure 2.3.: The Lagrangian specification of the solid motion. $\xi_{1,2,3}$ represent the origin coordinates

The measures of the deformation are the displacement and the (Lagrangian) strain tensors. The displacement is given by

$$\mathbf{u} = \mathbf{x} - \chi. \tag{2.11}$$

The vector $\mathbf{u}$ connects the undeformed reference configuration χ of a material particle to its deformed current configuration $\mathbf{x}$ at time t. The displacement is the measure of the change from χ to $\mathbf{x}$. The mapping between the reference and final state is carried out with the deformation gradient $\mathbf{F} = \partial\mathbf{x}/\partial\chi$ that represents the Jacobian matrix of motion.

On the other hand, the Lagrangian strain tensor, sometimes called the Green-Lagrange strain tensor, establishes a relationship between the displacements and the distortions. It is defined as,

$$\mathbf{E} = \frac{1}{2}\left(\mathbf{F}^T \cdot \mathbf{F} - \mathbf{I}\right). \tag{2.12}$$

We can interpret $\mathbf{E}$ as a measure for the change of length in a body.

2.3.2. Newton's second law of motion

The motion of the body is described by the instantaneous velocity field defined as $\partial\mathbf{u}/\partial t$ and obeys *Newton's second law of motion*. It states the balance of the internal forces related to stresses and the external forces applied.

In Lagrangian coordinates, Newton's second law can be written as

$$\rho\frac{\partial^2\mathbf{u}}{\partial t^2} = \nabla \cdot \left(J\sigma\mathbf{F}^{-T}\right) + \rho\mathbf{g}, \tag{2.13}$$

where ρ is the density of the structural material, $J = det\,(\mathbf{F})$ is the Jacobian determinant of the deformation gradient matrix, and σ is the Cauchy stress tensor field that depends on the material properties. The Cauchy stress tensor is also known as the *true stress* because it is defined in the reference's spatial state [44].

2.3.3. Constitutive law

The constitutive law, or stress-strain relation, describes the specific behavior of the material in response to the motion. It relates the deformation (strain) and internal forces (stress). The formulation of this relationship is based on physical observations of the material's response. When the stress is proportional to the strain, the material is a linear elastic material. Otherwise, it is a non-linear elastic material known as hyperelastic material [105]. A linear elastic material model can be used to represent the response of almost any material, e.g., conventional steels, some plastics, glass, and concrete. Mandatory prerequisites are that the strains and stresses are small, and the deformation occurs within the material's elastic range [185]. In contrast, a hyperelastic material model is useful for predicting the response of soft materials such as rubber and biological soft tissue.

The isotropic linear elastic material is characterized by an elastic potential that only retains the quadratic terms in the linearized strain tensor $\varepsilon = \frac{1}{2}\left(\nabla \mathbf{u} + \nabla \mathbf{u}^T\right)$. This constitutive law is written as

$$\sigma_s = 2\mu\varepsilon + \lambda tr(\varepsilon)\mathbf{I}. \tag{2.14}$$

λ and μ are the Lamé constants which depend on the material properties and are defined as

$$\lambda = \frac{\nu E}{(1+\nu)(1-2\nu)}, \quad \mu = \frac{E}{2(1+\nu)}, \tag{2.15}$$

where E is the module of elasticity and ν the Poisson's coefficient. Other useful elastic properties to define the material are the shear modulus G and bulk modulus K

$$G = \frac{E}{2(1+\nu)}, \quad K = \frac{E}{3(1-2\nu)}. \tag{2.16}$$

The most straightforward hyperelastic material model is the *Saint Venant–Kirchhoff model*. It is essentially an extension of the geometrically linear elastic material model to the geometrically nonlinear regime. The constitutive law is the same equation (2.14) but the strain tensor is the Lagrangian Green strain tensor $\mathbf{E}$ defined as

$$\mathbf{E} = \frac{1}{2}\left[\nabla \mathbf{u} + \nabla \mathbf{u}^T + \nabla \mathbf{u}^T \cdot \nabla \mathbf{u}\right]. \tag{2.17}$$

Besides, the stresses for a hyperelastic material can be expressed as derivatives of a strain energy density function W given by

$$W(\mathbf{E}) = \frac{\lambda}{2}[tr(\mathbf{E})]^2 + \mu tr\left(\mathbf{E}^2\right) \tag{2.18}$$

There are other more sophisticated hyperelastic models such as the neo-Hooke, Mooney-Rivlin, and Ogden. A good summary of them can be found in [44].

2.4. Boundary and interface coupling conditions

Finally, the mathematical modeling of the problem is closed by given suitable boundary and interface conditions. Various boundary conditions can be prescribed on the boundaries, such as periodic, slip, no-slip, inflow, and outflow on the two-fluid flow part. While on the structural part, specified displacements (Dirichlet condition) or specified surface tractions (von Neuman condition) must be assigned on the solid wall boundaries.

Furthermore, at the fluid-structure interface, both the dynamic and kinematic conservation need to be satisfied to ensure a correct energy balance at all times [56]. Thus, preventing the fluid and structural domains separate or overlap during movement. The continuity of the velocity (kinematic condition) and the equilibrium of stresses (dynamic condition) are fulfilled by setting

$$\mathbf{v}^{\mathrm{f}} = \frac{\partial \mathbf{u}^{s}}{\partial t} \quad \text{and} \quad \tau^{\mathrm{f}} \cdot \mathbf{n}^{\mathrm{fsi}} = \sigma^{\mathrm{s}} \cdot \mathbf{n}^{\mathrm{fsi}} \quad \text{on } \Gamma^{\mathrm{fsi}} \ , \tag{2.19}$$

where $\mathbf{n}^{\mathrm{fsi}}$ is the normal unit vector to the fluid-structure interface Γ^{fsi}.

3. Numerical solution of the Navier-Stokes equations

The numerical solution of the Navier-Stokes equations demands space and time discretization. The different discretization methodologies for each term of the Navier Stokes equations are well documented in numerous books such as [60, 61, 147, 169, 185, 230]. This chapter focuses only on the existing discretization techniques in the in-house FASTEST code and the new implementations to adapt it to solve two-fluid flows. Although the one-fluid formulation of the Navier-Stokes equations introduced in Chapter 2 includes an indicator function that identifies each fluid, its solution is not shown in this part but covered in the next chapter. The presentation is limited to incompressible laminar flows because the multifluid flows considered in this investigation all involve relatively low velocities and low Reynolds numbers.

In Section 3.1, the already existent main features of the in-house solver FASTEST are summarized. Section 3.2 shows the domain discretization, followed by the description of the interpolation schemes in Section 3.3. The momentum equation's discretization procedure is given in Section 3.4 and its resulting linear algebraic equation in Section 3.5. The accuracy of two-fluid flow computations requires employing a special interpolation technique to compute the viscosity and density at the cell-faces near the flow interface. So, Section 3.6 addresses this topic. Then, Section 3.7 outlines the continuity equation transformed into a pressure-correction equation. For clarity, the ***italic bold style*** highlights the new techniques implemented to solve multiphase flows. The chapter finishes with Section 3.8, showing the flowchart of a new pressure-velocity coupling algorithm that significantly improves the program's efficiency. This section includes the code verification of the new implementations by the method of manufactured solutions, a validation test case, and a study of the Rhie-Chow interpolation modifications.

3.1. The in-house solver FASTEST

The in-house FASTEST [46] code is a CFD solver written in Fortran programming language and designed to solve the Navier-Stokes equations governing the dynamics of incompressible 3D fluids in the laminar or turbulent regime. The solution algorithm relies on the finite volume method (FVM) for spatial discretization and the finite difference method (FD) for temporal discretization. The problem geometry is discretized with block-structured hexahedral collocated grids.

The following are some of the most relevant features of the solver:

- Fully implicit integration in time: implicit Euler, BDF2, and Crank-Nicolson method
- Flux-blending technique and high-order schemes to compute the convective term

- Derivative approximation based on the Coordinate Transformation scheme (CTS) and DABT scheme [114]
- Pressure-velocity coupling via the SIMPLE pressure-correction algorithm
- Boundary conditions implementation with ghost cells
- The solution of linear equation systems by the incomplete lower-upper decomposition (ILU)
- ALE formulation for moving grids including grid distortion interpolation technics: linear, transfinite, and elliptic
- Fully parallelized by grid partitioning
- Multigrid cycles for computing acceleration

3.2. Domain Discretization

Before starting the discretization procedure of the equations, the physical domain must be discretized. This process means subdividing the domain into discrete non-overlapping cells, commonly referred to as control volumes (CV), which completely cover the computational domain to form a grid or mesh system [147]. In FASTEST, a control volume is a hexahedron defined by eight vertices (1-8) and a center point P, as illustrated in Figure 3.1. The surrounding CV-centers are identified by W, E, S, N, B, and T. They refer to west, east, south, north, bottom, and top. The designation corresponds to the orientation of the adjacent centers for point P. The CV-faces centers are identified by corresponding lower case letters: w, e, s, n, b, and t.

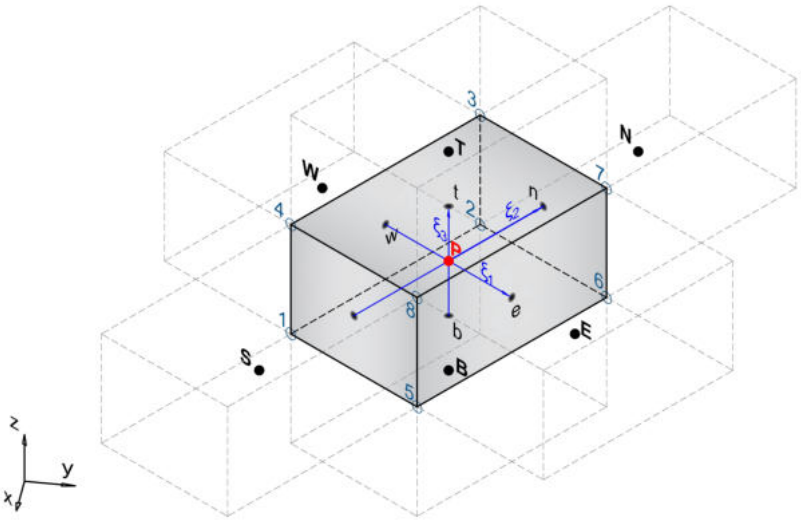


Figure 3.1.: The topology of a control volume in FASTEST with a local coordinate system associated with the cell center

The finite volume method in FASTEST is developed for use in conjunction with Cartesian grids. Therefore, the irregular physical domain defined by the coordinates $\mathbf{x} = (x, y, z)$ is transformed internally into a Cartesian computational domain with coordinates $\xi = (\xi_1, \xi_2, \xi_3)$ by applying the coordinate transformation scheme (CTS) described in [60, 61, 114, 148]. The local Cartesian coordinates are associated with each control volume. The transformation matrix $\mathbf{B}_{ij}$ characterizes the mapping between the two coordinate systems and is given by

$$\mathbf{B}_{ij} = \left(\frac{\partial x_i}{\partial \xi_j}\right) = \begin{bmatrix} \frac{\partial x}{\partial \xi_1} & \frac{\partial x}{\partial \xi_2} & \frac{\partial x}{\partial \xi_3} \\ \frac{\partial y}{\partial \xi_1} & \frac{\partial y}{\partial \xi_2} & \frac{\partial y}{\partial \xi_3} \\ \frac{\partial z}{\partial \xi_1} & \frac{\partial z}{\partial \xi_2} & \frac{\partial z}{\partial \xi_3} \end{bmatrix}, \quad i, j = 1, 2, 3. \tag{3.1}$$

The matrix components are computed with the second-order central differences method. Generally, $\Delta\xi_j = 1$ can be assumed [60], and the coordinates of the face centers are found in terms of the eight CV-vertices.

The use of CTS to define a regular computational domain implies in the discretization of the equations the transformation of the derivatives in the global coordinates system x_i into derivatives in the local Cartesian coordinates system ξ_j

$$\frac{\partial \phi}{\partial x_i} = \frac{\partial \phi}{\partial \xi_j}\frac{\partial \xi_j}{\partial x_i} = \frac{\partial \phi}{\partial \xi_j}\mathbf{B}_{ij}^{-1} = \frac{\partial \phi}{\partial \xi_j}\frac{\beta_{ij}}{J}, \tag{3.2}$$

where $J = \det(\mathbf{B}_{ij})$ is the Jacobian and $\beta_{ij} = adj(\mathbf{B}_{ij})$ is the adjoint matrix of $\mathbf{B}_{ij}$.

3.3. Interpolation

The discretization of some terms of the equations requires the face center values of the variables. Because the code has a collocated variables arrangement, they derive from the interpolation of the central nodal values. The available interpolation methods are the well-known linear interpolation or also called the central differencing scheme (CDS), and the multidimensional linear interpolation (MuLI) developed by Lehnhäuser and Schäfer [115].

The **linear interpolation** approximates the CV-face value in terms of the two nearest nodes. At face e, we have

$$\phi_e = \lambda_E \phi_E + (1 - \lambda_E)\,\phi_P. \tag{3.3}$$

The **MuLI** calculates the face value using the Taylor series expansions of the six nodal points near the face center in the three local directions. So, at face e, the value results from weighting the nodal points P, E, N, S, T, and B by applying the following expression

$$\phi_e = \lambda_E \phi_E + \lambda_P \phi_P + \lambda_{NS}\,(\phi_N - \phi_S) + \lambda_{TB}\,(\phi_T - \phi_B)\,. \tag{3.4}$$

The λ factors are geometric factors whose definition is presented in [114].

The linear interpolation provides second-order accuracy on uniform Cartesian grids and formal first-order accuracy on non-uniform grids [60]. In contrast, the MuLI scheme has a second-order

accuracy independent of any grid distortion because it preserves the control volume sparsity pattern [115].

3.4. Discretization of the momentum equation

The FVM discretization process starts by integrating the momentum equation, Eq. (2.4), over each control volume of the computational domain that enables recovering its integral form. Then, applying the Gauss theorem, the volume integrals are transformed into surface integrals. For the control volume P, it yields to

$$\underbrace{\rho\frac{\partial}{\partial t}\int\limits_{V_P}\mathbf{v}dV}_{\text{transient term}}+\underbrace{\int\limits_{S_p}\rho\left(\mathbf{v}-\mathbf{v}_{\mathrm{g}}\right)\mathbf{v}\cdot\mathbf{n}dS}_{\text{convection term}}=\underbrace{-\int\limits_{V_P}\left(\nabla p\right)dV}_{\text{pressure term}}+\underbrace{\int\limits_{S_p}\mu\left[\nabla\mathbf{v}+\left(\nabla\mathbf{v}\right)^{T}\right]\cdot\mathbf{n}dS}_{\text{diffusion term}}+\underbrace{\int\limits_{V_P}\left(\rho\mathbf{g}+\sigma\kappa\mathbf{n}^{\mathrm{fi}}\left|\nabla\alpha\right|\right)dV}_{\text{Body forces term}}. \tag{3.5}$$

$\mathbf{v}=(u,v,w)$ and $\mathbf{n}$ is the outward pointing unit vector normal to the surface S that closes the volume V of a control volume P.

3.4.1. The convection term

Replacing the surface integral by a summation of fluxes over the cell faces of the control volume P shown in Figure 3.1, the convection term results in

$$\int\limits_{S_P}\rho\left(\mathbf{v}-\mathbf{v}_{\mathrm{g}}\right)\mathbf{v}\cdot\mathbf{n}dS=\sum_{f(P)}\left[\underbrace{\rho_f\mathbf{v}_f\cdot\mathbf{n}\delta S_f}_{\dot{m}_f}-\underbrace{\rho_f\mathbf{v}_{\mathrm{g},f}\cdot\mathbf{n}\delta S_f}_{\dot{m}_{g,f}}\right]\mathbf{v}_f=\sum_{f(P)}F_f^{\mathrm{C}}\mathbf{v}_f, \tag{3.6}$$

where $f\left(P\right)=e,w,n,s,b,t$ represents each face of the control volume P, $\dot{m}_f$ is the mass flux, and $\dot{m}_{g,f}$ the grid flux trough the face f. Both fluxes form the total mass flow rate at the face denoted by F_f^C , and $\mathbf{v}_f$ is the interpolated face velocity vector.

The advection flux through the face $F_f^C\mathbf{v}_f$ is approximated by the flux-blending [185] technique given by

$$F_f^{\mathrm{C}}\mathbf{v}_f=\left(F_f^{\mathrm{C}}\mathbf{v}_f\right)^{\mathrm{LO}}+\underbrace{\psi^{\mathrm{C}}\left[\left(F_f^{\mathrm{C}}\mathbf{v}_f\right)^{\mathrm{HO}}-\left(F_f^{\mathrm{C}}\mathbf{v}_f\right)^{\mathrm{LO}}\right]}_{\text{Explicit convection source }\mathbf{S}_f^{\mathrm{C}}}. \tag{3.7}$$

It blends the accurate approximation of the higher-order (HO) schemes and the better robustness and boundedness properties of the lower order (LO) schemes. The blending factor $0\leq\psi^C\leq1$ defines the influence of the HO scheme in the final value. The difference between the HO and LO values is explicitly computed and represents a source term $\mathbf{S}_f^C$.

The LO interpolation scheme is the first-order upwind differencing scheme (UDS) that defines $F_f^C \mathbf{v}_f$ depending on the convection flux's direction. Thus, for example, the advection flux across the east face is:

$$\left(F_e^{\mathrm{C}} \mathbf{v}_e\right)^{\mathrm{UDS}} = \min\left(F_e^{\mathrm{C}}, 0\right) \mathbf{v}_E + \max\left(F_e^{\mathrm{C}}, 0\right) \mathbf{v}_P. \tag{3.8}$$

The UDS imitates the advection's fundamental physics, so it is an unconditionally stable and bounded scheme. However, it introduces artificial numerical diffusion.

In contrast, the HO interpolation scheme can be the second-order CDS scheme, the MuLI scheme, or a HO scheme as the QUICK [116] scheme (more information about the HO schemes available in FASTEST is given in [170]). For instance, for the face e, the advection flux using MuLI results in

$$\left(F_e^{\mathrm{C}} \mathbf{v}_e\right)^{\mathrm{MuLI}} = F_e^{\mathrm{C}} \left[\lambda_E \mathbf{v}_E + \lambda_P \mathbf{v}_P + \lambda_{NS} \left(\mathbf{v}_N - \mathbf{v}_S\right) + \lambda_{TB} \left(\mathbf{v}_T - \mathbf{v}_B\right)\right]. \tag{3.9}$$

3.4.2. The diffusion term

The diffusion term discretization requires approximating the normal derivative of the velocity in the center of the CV-faces. For Cartesian grids, these derivatives can be approximated using a central difference formula. However, for distorted grids, which are regularly non-orthogonal, the face center is not part of the line between the nodal points near the face, which can cause oscillatory solutions [60, 185]. In these cases, the derivative approximation based on the multidimensional Taylor series expansion (DABT) scheme developed in [114] is applied (see Appendix A for further explanation of the DABT scheme).

Applying the DABT scheme to approximate the derivatives of the velocity, the discretized diffusion term resulting in

$$\begin{aligned}
\int_{S_p} \mu \left[\nabla \mathbf{v} + (\nabla \mathbf{v})^T\right] \cdot \mathbf{n} dS &= \sum_{f(P)} \frac{\mu_f \delta S_f}{J^{f,DABT}} \left(\beta_{ij}^{f,DABT} \cdot \mathbf{n}\right) \cdot (\Delta \mathbf{v})_{f,DABT}^{ij} \\
&+ \sum_{f(P)} \frac{\mu_f \delta S_f}{J^{f,DABT}} \left[\left(\beta_{ij}^{f,DABT}\right)^T \cdot (\Delta \mathbf{v})_{f,DABT}^{ij}\right] \cdot \mathbf{n} \\
&= \sum_{f(P)} \mathbf{F}_f^{\mathrm{D}}
\end{aligned} \tag{3.10}$$

Let $\mathbf{F}_f^D$ be the resulting diffusion flux term through the face consisting of an implicit and explicit part.

For clarifying, the diffusion flux through the face e is exemplified. We consider the matrix $\mathbf{B}_{ij}^{e,DABT}$ defined by Eq. (A.2) and the multidimensional variation of the velocity vector around the face e given as

$$\begin{aligned}
(\Delta \mathbf{v})_{e,DABT}^{1j} &= (v_{j,E} - v_{j,P}) \\
(\Delta \mathbf{v})_{e,DABT}^{2j} &= (v_{j,N} - v_{j,S} + v_{j,NE} - v_{j,SE}) \\
(\Delta \mathbf{v})_{e,DABT}^{3j} &= (v_{j,T} - v_{j,B} + v_{j,TE} - v_{j,BE})
\end{aligned} \tag{3.11}$$

where $j = 1, 2, 3$ and denotes the three velocity components.

Hence, the diffusion flux through the face e is:

$$\begin{aligned}\mathbf{F}_e^{\mathrm{D}} = & \frac{\mu_e \delta S_e}{J^{e,DABT}} \left(\beta_{1j}^{e,DABT} \cdot \mathbf{n}\right) (\mathbf{v}_E - \mathbf{v}_P) \\ & + \underbrace{\frac{\mu_e \delta S_e}{J^{e,DABT}} \left[\left(\beta_{2j}^{e,DABT} \cdot \mathbf{n}\right) (\mathbf{v}_N - \mathbf{v}_S + \mathbf{v}_{NE} - \mathbf{v}_{SE}) + \left(\beta_{3j}^{e,DABT} \cdot \mathbf{n}\right) (\mathbf{v}_T - \mathbf{v}_V + \mathbf{v}_{TE} - \mathbf{v}_{SE})\right]}_{\text{Explicit diffusion source } \mathbf{S}_e^{\mathrm{D1}}} \\ & + \underbrace{\frac{\mu_e \delta S_e}{J^{e,DABT}} \left[\left(\beta_{ij}^{e,DABT}\right)^T \cdot (\Delta \mathbf{v})_{e,DABT}^{ij}\right] \cdot \mathbf{n}}_{\text{Explicit diffusion source } \mathbf{S}_e^{\mathrm{D2}}}.\end{aligned} \tag{3.12}$$

The first term involving the CVs' nodal velocities that form the face is the implicit part. In contrast, the terms containing the surrounding velocities to the face and the transposed matrix of $\beta_{ij}^{e,DABT}$ are the explicit diffusion sources $\mathbf{S}_e^{D1}$ and $\mathbf{S}_e^{D2}$. The explicit parts vanish for orthogonal grids and fluids with constant viscosity [60].

3.4.3. The pressure term

Unlike the diffusive term, the pressure gradient is approximated with the CTS scheme with a local axis associated with the CV-center P pictured in Figure 3.1. Consequently, employing Eq. (3.2), the discretization of the pressure term for the control volume P outcomes

$$\int_{V_P} (\nabla p)\, dV = \frac{\partial p}{\partial \xi_j} \frac{\beta_{ij}^P}{J^P} \delta V_P = (\Delta p)_P^j \cdot \beta_{ij}^P. \tag{3.13}$$

J^P represents the volume of cell P, which simplifies δV_P of the final expression. The adjoint matrix β_{ij}^P and the Jacobian determinant J^P refer to the transformation matrix $\mathbf{B}_{ij}^P$ given by Eq. (3.1). The local pressure gradient is determined as

$$(\Delta p)_P^j = \frac{\partial p}{\partial \xi_j} = \begin{pmatrix} p_e - p_w \\ p_n - p_s \\ p_t - p_b \end{pmatrix}. \tag{3.14}$$

The partial derivatives of p with respect to the local axis ξ_j are central difference approximations considering $\Delta \xi_j = 1$. The face quantities result from linear or MuLI interpolation.

3.4.4. The body forces term

A logical idea for the discretization of the body forces term is to apply the midpoint rule, which would result in

$$\int_{V_P} \left(\rho \mathbf{g} + \sigma \kappa \mathbf{n}^{fi} |\nabla \alpha|\right) dV = \left(\mathbf{g} \rho_P + \sigma \kappa_P \nabla \alpha_P\right) \delta V_P = \mathbf{b}_P \delta V_P. \tag{3.15}$$

$\mathbf{b}_P$ represents the total mean body forces at the center of the control volume P.

However, this simple estimation of the body forces term cannot be used directly to discretize the one-fluid formulation of the momentum equation. Because in multiphase flows (e.g., air-water), the free surface appears as a density discontinuity. It produces a sudden variation in body forces, which induces abrupt variations of the pressure and introduces nonphysical spikes in the velocity field near the interface [143].

The relationship between abrupt changes of body forces and abrupt variation in pressure becomes evident if we consider the momentum equation for a quiescent fluid ($\mathbf{v} = 0$):

$$0 = -\nabla p + \underbrace{\left(\rho \mathbf{g} + \sigma \kappa \mathbf{n}^{fi} \left|\nabla \alpha\right|\right)}_{\mathbf{b}}. \tag{3.16}$$

A zero velocity field implies a counterbalance between the discretized pressure gradient and body force. This balance prevents parasitic currents due to gravity, even when surface tension is neglected [224]. A wrong balance leads to unphysical high velocities in the free surface cells containing the lighter fluid [167].

In the context of this investigation, ***the approach introduced by Mencinger [142] for the collocated variable arrangement is implemented to minimize parasitic currents***. This method consists of a redistribution of the body force term to evaluate the body forces in a similar stencil as the pressure gradient to ensure the equation balance. This redistribution can best be derived by considering the one-dimensional situation illustrated in Figure 3.2. The double bar indicates two average steps, and F any neighboring CV.

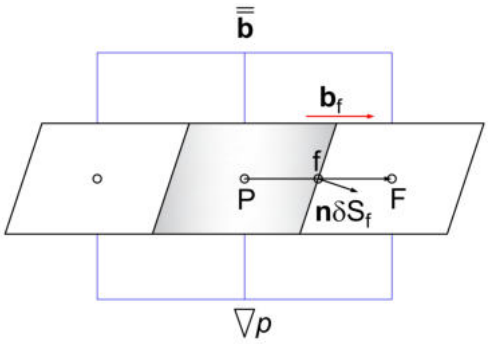


Figure 3.2.: One-dimensional stationary flow in a distorted mesh

The method assumes that the solution of the momentum equation for a quiescent fluid contained in the control volume P given by

$$(\nabla p)_P = \mathbf{b}_P, \tag{3.17}$$

is also satisfied on all control volume faces. Thus for the face f,

$$(\nabla p)_f = \mathbf{b}_f. \tag{3.18}$$

The pressure gradient at the CV-center P is determined with the Green Gauss gradient equation,

$$(\nabla p)_P = \frac{1}{\delta V_P} \sum_{f(P)} p_f \mathbf{n} \delta S_f = \frac{1}{\delta V_P} \sum_{f(P)} \left(\lambda_F p_F + (1-\lambda_F)\, p_P\right) \mathbf{n} \delta S_f, \tag{3.19}$$

where the face pressure value is expressed as a linear interpolation of the CV-centers.

On the other hand, the pressure gradient at the CV-face f, Eq. (3.18), is computed with central differences, which results in the following expression for the neighboring nodal pressure p_F

$$p_F = p_P + \mathbf{d}_{PF} \cdot \mathbf{b}_f. \tag{3.20}$$

The vector $\mathbf{d}_{PF}$ connects the cell centers P and F.

Substituting Eq. (3.20) in Eq. (3.19) and after simplifications, it becomes

$$\begin{aligned} (\nabla p)_P &= \frac{1}{\delta V_P} \sum_{f(P)} (P_P + \lambda_F \mathbf{d}_{PF} \cdot \mathbf{b}_f) \mathbf{n} \delta S_f \\ &= \frac{P_P}{\delta V_P} \sum_{f(P)} \mathbf{n} \delta S_f + \frac{1}{\delta V_P} \sum_{f(P)} (\lambda_F \mathbf{d}_{PF} \cdot \mathbf{b}_f) \mathbf{n} \delta S_f, \end{aligned} \tag{3.21}$$

where the first term vanishes because $\sum_{f(P)} \mathbf{n} \delta S_f = 0$ for any closed surface.

Replacing Eq. (3.21) in the initial equilibrium statement Eq. (3.17), the re-distributed body forces $\overline{\overline{\mathbf{b}}}_P$ term results in

$$\overline{\overline{\mathbf{b}}}_P = (\nabla p)_P = \frac{1}{\delta V_P} \sum_{f(P)} \lambda_F (\mathbf{d}_{PF} \cdot \mathbf{b}_f) \mathbf{n} \delta S_f. \tag{3.22}$$

The above expression must be used in the momentum equation instead of the mean body forces to avoid unphysical spikes in the velocity field.

Because the code is developed for local Cartesian coordinates, the re-distributed body force still requires a transformation before implementation. Assuming that

$$\lambda_F (\mathbf{d}_{PF} \cdot \mathbf{b}_f) = \varphi_f, \tag{3.23}$$

Eq. (3.22) is rewritten as

$$\overline{\overline{\mathbf{b}}}_P = \frac{1}{\delta V_P} \sum_{f(P)} \varphi_f \mathbf{n} \delta S_f = \left(\frac{\partial \varphi}{\partial x_i} \right)_P. \tag{3.24}$$

The right-hand side of Eq. (3.24) is the expression of the Green Gauss gradient of the variable φ at the centroid of the cell P. Hence, $\overline{\overline{\mathbf{b}}}_P$ is equal to the gradient of the variable φ, which expressed in local coordinates with the CTS method, Eq. (3.2), and using the same transformation matrix defined for the pressure term, Eq. (3.1), becomes

$$\overline{\overline{\mathbf{b}}}_P = \frac{\partial \varphi}{\partial \xi_j} \frac{\beta_{ij}^P}{J^P} = (\Delta \varphi)_P^j \cdot \frac{\beta_{ij}^P}{J^P}, \tag{3.25}$$

where the local gradient of φ is expressed as

$$(\Delta\varphi)_P^j = \frac{\partial\varphi}{\partial\xi_j} = \begin{pmatrix} \varphi_e - \varphi_w \\ \varphi_n - \varphi_s \\ \varphi_t - \varphi_b \end{pmatrix} = \begin{bmatrix} \lambda_E(\mathbf{d}_{PE}\cdot\mathbf{b}_e) - \lambda_W(\mathbf{d}_{PW}\cdot\mathbf{b}_w) \\ \lambda_N(\mathbf{d}_{PN}\cdot\mathbf{b}_n) - \lambda_S(\mathbf{d}_{PS}\cdot\mathbf{b}_s) \\ \lambda_T(\mathbf{d}_{PT}\cdot\mathbf{b}_t) - \lambda_B(\mathbf{d}_{PB}\cdot\mathbf{b}_b) \end{bmatrix}. \tag{3.26}$$

The body forces at the CV-faces are linear or multidimensional linear interpolations of the mean nodal values given by

$$\mathbf{b}_P = (\mathbf{g}\rho_P + \sigma\kappa_P\nabla\alpha_P)\,. \tag{3.27}$$

The surface tension force in the cell center can be computed with any available surface tension models of the FASTEST solver. They are discussed in detail in [189] and the new implementations in [63].

Finally, using the re-distributed body force given by Eq. (3.25), the suitable discretization of the body forces term is written as

$$\int_{V_P} \left(\rho\mathbf{g} + \sigma\kappa\mathbf{n}^{fi}\,|\nabla\alpha|\right) dV = (\Delta\varphi)_P^j \cdot \beta_{ij}^P. \tag{3.28}$$

3.4.5. The transient term

The temporal discretization with the finite difference (FD) method requires a time coordinate along which the time derivative can be evaluated step by step [147]. First, the time interval $[t_0, T]$ under consideration is divided into small time intervals Δt_n,

$$t_{n+1} = t_n + \Delta t_n, \quad n = 0, 1, 2, 3, ... \tag{3.29}$$

Then, starting with an initial condition at time $t = t_0$. The solution method advances and finds the solution at the time $t_1 = t_0 + \Delta t_0$. The solution at t_1 is the new initial condition and is used to obtain the solution at the next time $t_2 = t_1 + \Delta t_1$. In the same way, the process is repeated until the solution at the last time is found. Due to unsteady flows are parabolic in time, the solution at t_{n+1} only depends on the solutions at earlier times [59].

FASTEST has available for the temporal discretization of moving domain problems two fully implicit variants of the FD approximations:

- the first-order forward Euler method

$$\rho_P^{n+1}\frac{(\mathbf{v}_P\delta V_P)^{n+1} - (\mathbf{v}_P\delta V_P)^n}{\Delta t_n} = L\left({\mathbf{v}_P}^{n+1}\right), \tag{3.30}$$

- and the backward-differencing formula of second-order (BDF2)

$$\rho_P^{n+1}\frac{3(\mathbf{v}_P\delta V_P)^{n+1} - 4(\mathbf{v}_P\delta V_P)^n + (\mathbf{v}_P\delta V_P)^{n-1}}{2\Delta t_n} = L\left({\mathbf{v}_P}^{n+1}\right), \tag{3.31}$$

The superscripts $n+1$, n, and $n-1$ denote the variable value at time t_{n+1}, t_n, and t_{n-1}. The $L\left(\mathbf{v}_P^t\right)$ term is the spatial discretization operator that includes all non-transient terms (e.g., convection,

advection, body force, pressure, and source terms) evaluated at a reference time t. Besides note, the density is outside the derivative approximation because it is considered constant during the time interval and equal to the last update.

One crucial requirement for multifluid flows is to maintain low numerical diffusion and not introduce spurious oscillations near the sharp interface [94]. This requirement is generally satisfied by keeping the Courant (Co) number smaller than one. The Courant number is a measure of how much information traverses a computational grid cell in a given time-step. A common practice to control the Co number is a variable time-step (see e.g., [101]). For the first order implicit Euler scheme, the discretization is not affected by whether the time-step is variable or constant. However, for BDF2, since it involves two time-steps, the interpolation profile must be modified to account for the non-uniform time intervals.

Following the approach presented in [147], in this work, ***the BDF2 scheme is modified to use variable time-steps in the case of multiphase flows***, which results in

$$\rho^{n+1}\left[k_P^{n+1}(\mathbf{v}_P\delta V_P)^{n+1}+k_P^{n}(\mathbf{v}_P\delta V_P)^{n}+k_P^{n-1}(\mathbf{v}_P\delta V_P)^{n-1}\right]=L\left({\mathbf{v}_P}^{n+1}\right), \tag{3.32}$$

where the time dependent coefficients are

$$\begin{aligned} k_P^{n+1} &= \left(\frac{1}{\Delta t_n}+\frac{1}{\Delta t_n+\Delta t_{n-1}}\right), \\ k_P^{n} &= -\left(\frac{1}{\Delta t_n}+\frac{1}{\Delta t_{n-1}}\right), \\ k_P^{n-1} &= \left(\frac{\Delta t_n}{\Delta t_{n-1}\left(\Delta t_n+\Delta t_{n-1}\right)}\right). \end{aligned} \tag{3.33}$$

For uniform time-steps, the coefficients given in Eq. (3.31) are recovered.

3.5. The discretized momentum equation

Using the spatial discretization expressions developed in the previous section for the centroid of a control volume P with neighboring CVs F and applying the BDF2 for the temporal discretization with variable time-steps, the discretized momentum equation using vector notation is written as

$$A_P^{\mathbf{v}}\mathbf{v}_P^{n+1}=\sum_{F(P)}A_F^{\mathbf{v}}\mathbf{v}_F^{n+1}+\mathbf{S}_P^{\mathbf{v},n}, \tag{3.34}$$

with the coefficients calculated using the following statements:

$$
\begin{aligned}
A_F^{\mathbf{v}} &= -\min\left(F_f^{\mathrm{C},m}, 0\right) + \left(\frac{\mu_f \delta S_f}{J^{f,DABT}} \beta_{1j}^{f,DABT} \cdot \mathbf{n}\right)^{n+1}, \\
A_P^{\mathbf{v}} &= \sum_{F(P)} A_F^{\mathbf{v}} + k_P^{n+1} \left(\rho \delta V\right)_P^{n+1}, \\
\mathbf{S}_P^{\mathbf{v},n} &= \sum_{f(P)} \underbrace{\psi^C \left[\left(F_f^C \mathbf{v}_f\right)^{HO} - \left(F_f^C \mathbf{v}_f\right)^{UDS}\right]^m}_{\text{Convection source term } \left(\mathbf{s}_f^{\mathrm{C}}\right)^{\mathbf{v},m}} \\
&\quad + \sum_{f(P)} \underbrace{\left(\frac{\mu_f \delta S_f}{J^{f,DABT}}\right)^{n+1} \left[\begin{array}{l} \left(\beta_{2j}^{f,DABT} \cdot \mathbf{n}\right)^{n+1} \left(\mathbf{v}_{N(f)}^m - \mathbf{v}_{S(f)}^m + \mathbf{v}_{NE(f)}^m - \mathbf{v}_{SE(f)}^m\right) \\ + \left(\beta_{3j}^{f,DABT} \cdot \mathbf{n}\right)^{n+1} \left(\mathbf{v}_{T(f)}^m - \mathbf{v}_{B(f)}^m + \mathbf{v}_{TE(f)}^m - \mathbf{v}_{BE(f)}^m\right) \end{array}\right]}_{\text{Diffusion source term } \left(\mathbf{s}_f^{\mathrm{D1}}\right)^{\mathbf{v},m}} \\
&\quad + \sum_{f(P)} \underbrace{\left(\frac{\mu_f \delta S_f}{J^{f,DABT}}\right)^{n+1} \left(\left(\beta_{ij}^{f,DABT}\right)^{T,n+1} \cdot \left(\Delta \mathbf{v}\right)_f^{j,m}\right) \cdot \mathbf{n}^{n+1}}_{\text{Diffusion source term } \left(\mathbf{s}_f^{\mathrm{D2}}\right)^{\mathbf{v},m}} \\
&\quad - \underbrace{\left(\Delta p\right)_P^{j,m} \cdot \beta_{ij}^{P,n+1}}_{\text{Pressure term}} + \underbrace{\left(\Delta_j \varphi\right)_P^m \cdot \beta_{ij}^{P,n+1}}_{\text{Body forces term}} + \underbrace{\rho_P^{n+1} \left[k_P^n (\mathbf{v}_P \delta V_P)^n - k_P^{n-1} (\mathbf{v}_P \delta V_P)^{n-1}\right]}_{\text{Temporal source term}},
\end{aligned}
\tag{3.35}
$$

The superscript m represents a value evaluated in a previous iteration.

An iterative process proposed by [158] allows Eq. (3.34) to be solved as a linear equation, although its coefficients depend on the nonlinear velocity and pressure fields. The coefficients are recomputed in each iteration based on the values obtained in the previous iteration. Furthermore, the momentum equation is implicitly under-relaxed with a parameter $0 < \gamma^{\mathbf{v}} < 1$ to minimize the effects of the continuous variation of the coefficient values. This variation can cause significant changes in $\mathbf{v}$ deteriorating the convergence rate or often leading to divergence [185]. The under-relaxed momentum equation is expressed as

$$
\underbrace{\frac{A_P^{\mathbf{v}}}{\gamma^{\mathbf{v}}}}_{\tilde{A}_P^{\mathbf{v}}} \mathbf{v}_P^{n+1} = \sum_{f(P)} A_F^{\mathbf{v}} \mathbf{v}_F^{n+1} + \underbrace{\mathbf{S}_P^{\mathbf{v},n} + \frac{1-\gamma^{\mathbf{v}}}{\gamma^{\mathbf{v}}} A_P^{\mathbf{v}} \mathbf{v}_P^m}_{mathbf{S}_P^{\mathbf{v},n}},
\tag{3.36}
$$

which can adopt the form of Eq. (3.34) again, if the coefficient $A_P^{\mathbf{v}} = \tilde{A}_P^{\mathbf{v}}$ and $\mathbf{S}_P^{\mathbf{v},n} = \tilde{\mathbf{S}}_P^{\mathbf{v},n}$.

Equation (3.34) applies to all control volumes that are not located at the boundaries of the problem domain. For boundary control volumes, the coefficients are modified according to the given type of boundary conditions. Resulting finally in a linear system of N equations, which is solved via the incomplete LU decomposition method of Stone [192].

3.5.1. The boundary conditions

The boundary conditions are implemented at the borders of the computational domain using ghost cells. In the FASTEST solver, three physical boundary conditions: *inlet, outlet*, and *no-slip wall*, and two geometric constraints: *symmetry* and *periodicity*, are available.

a. At an *inlet* boundary, the velocity vector is prescribed. The convection and diffusion fluxes at the face boundary are calculated using the known velocity.

b. At an *outlet* boundary, a fully developed flow is assumed. The normal velocity gradient is zero, and the velocity on the face boundary is equal to the nearest central interior value.

c. A *no-slip* wall boundary means that the fluid is in contact with a solid wall, and the boundary velocity is equal to the wall velocity. It implies that the normal stress at the wall is zero and that the exchange of momentum is transmitted only by the wall shear stress.

d. A *symmetry* plane splits the computational domain into two parts that reflect each other when the variables' solution is symmetrical. In a symmetry boundary, the velocity component normal to it is zero while the component tangent to it keeps its magnitude and direction. The shear stress tangent to the plane is zero while the normal stress is not.

e. The *periodic* boundary reduces the computational domain when solutions are periodic in space. A pair of boundary surfaces are necessary to define the periodic direction. For instance, a periodic boundary condition can connect the left boundary of the domain to the right boundary or the top to the bottom. Each face in a periodic boundary has a matching face and neighbor cells on the other periodic boundary applying a geometric transformation. Hence, the boundary CVs are treated as internal CVs, but the neighboring CVs are the other border's corresponding CVs.

The derivation of boundary conditions using the FVM can be found in [162] or more modern [147].

3.6. Face values of viscosity and density

Throughout the discretization process, the values of the viscosity and density on the CV-faces are required. However, these values are unknown and must be computed in terms of the known CV-centers values. For flows with uniform physical properties in the whole domain, the interpolation technique to obtain the face values is irrelevant. However, for flows with non-homogeneous physical properties, proper interpolation is essential to obtain a correct estimation of the flux through the faces. This issue was first addressed by Patankar [158], who proposed harmonic interpolation as an alternative to linear interpolation for dealing with non-uniform conductivity materials.

In the context of free surface flows with abrupt changes of the viscosity across the interface, references [30] and [203] demonstrated a proper interpolation of viscosity at the interface is crucial for reproducing the correct free surface dynamics. Though choosing between linear or harmonic interpolation is not straightforward for multifluid flows, the harmonic mean is accurate

when the interface tends to align with the flow direction. However, when it aligns with the CV-faces, the linear interpolation becomes exact. In practice, according to [178], harmonic interpolation is more accurate than linear interpolation, but it can be less robust for multiphase flows with a large density ratio. In contrast, the linear interpolation offers numerical stability, but it causes an artificial acceleration of the lighter phase's fluid elements, which yields too high velocities in low-density regions. Therefore, a combination of the two techniques is desirable.

Here, the interpolation issue is addressed by applying the approach presented in [110] and used in [136]: ***A combination of linear and harmonic interpolation weighted by the angle formed between the flow interface and the cell face***. Although the most critical problem in the literature is related to the face viscosity in the diffuse term, throughout this investigation, it was also observed that applying more accurate interpolation for the density improves the definition of the flow interface. Consequently, the approach is used to compute the face values of both viscosity and density.

Then, the value of a physical property $\phi = \rho, \mu$ at the face f formed between the control volume P and a neighbor F is computed as

$$\phi_f = \eta_f \phi_f^l + (1 - \eta_f)\, \phi_f^h \tag{3.37}$$

$$\text{with} \quad \phi_f^l = \lambda_F \phi_F + (1 - \lambda_F)\, \phi_P \quad \text{and} \quad \phi_f^h = \frac{\phi_F \phi_P}{\lambda_F \phi_F + (1 - \lambda_F)\, \phi_P}. \tag{3.38}$$

The superscripts l and h denotes linear and harmonic interpolation, respectively. In multifluid flows with large properties differences, linear interpolation yields face values closer to the higher nodal value, while harmonic interpolation to the lower one [110]. The factor $0 \leq n_f \leq 1$ weighs them depending on the interface orientation and the cell face. It is computed as

$$\eta_f = \left| \left(\frac{\nabla \alpha}{\|\nabla \alpha\|} \right)_f \cdot \frac{\mathbf{n} \delta S_f}{\|\mathbf{n} \delta S_f\|} \right| = \frac{1}{J^{f,DABT} \|\nabla \alpha\|_f} \left[(\Delta \alpha)_f^j \cdot \beta_{ij}^{f,DABT} \right] \cdot \mathbf{n}. \tag{3.39}$$

The volume fraction gradient on the face is approximated with the DABT method explained in Appendix A.

3.7. The discretization of the mass conservation equation

In the segregated pressure-based SIMPLE (Semi Implicit Method for Pressure Linked Equation) [158] algorithm adopted by the FASTEST solver, the mass equation is not solved directly. Instead, the mass equation is transformed into a pressure-correction equation by combining it with the momentum equation's algebraic form to obtain pressure and velocity fields that satisfy both equations.

The transformation starts by rewriting the under-relaxed momentum equation (3.36) in a more compact form

$$\mathbf{v}_P = \mathbf{H}_P \left[\mathbf{v} \right] + \widehat{\mathbf{S}}_P^{\mathbf{v}} - \mathbf{D}_P^{\mathbf{v}} \left(\nabla p_P \right), \tag{3.40}$$

with the following vector operators:

$$\mathbf{H}_p\left[\mathbf{v}\right] = \frac{\sum\limits_{f(P)} A_F^{\mathbf{v}} \mathbf{v}_F}{A_P^{\mathbf{v}}},\ \widehat{\mathbf{S}}_P^{\mathbf{v}} = \frac{\mathbf{S}_P^{\mathbf{v}} - (-\nabla p_P)}{A_P^{\mathbf{v}}},\ \mathbf{D}_P^{\mathbf{v}} = \frac{\delta V c}{A_P^{\mathbf{v}}}. \tag{3.41}$$

The source term $\widehat{\mathbf{S}}_P^{\mathbf{v}}$ contains all the source terms of the momentum equation except the pressure term. It is written as an individual term. The subscripts corresponding to the time discretization $(n, n+1)$ are omitted for simplicity.

The second step is to discretize the mass conservation equation, Eq. (2.3), employing the FVM and the divergence theorem, which yields

$$\sum_{f \sim nb(P)} \underbrace{\left(\rho \mathbf{v} \cdot \mathbf{n} \delta S\right)_f}_{\dot{m}_f} = 0, \tag{3.42}$$

where the mass flux $\dot{m}_f$ depends on the velocity at the face computed with the Rhie-Chow interpolation [171] as

$$\mathbf{v}_f = \mathbf{v}_f - \overline{\mathbf{D}_f^{\mathbf{v}}} \left(\nabla p_f - \overline{\nabla p_f}\right). \tag{3.43}$$

This expression decreases the pressure and velocity decoupling chances at the cell level, responsible for the checkboard pressure problem in collocated formulations [162, 172]. An overbar denotes linearly interpolated values.

The third step starts noting that the resulting velocity field $\mathbf{v}$ from solving Eq. (3.40) in the whole domain should be the exact solution that satisfies the momentum and mass conservation equation. However, since the velocity field is iteratively computed based on the pressure field from the previous iterations m, only the momentum conservation velocity field $\mathbf{v}^*$ is obtained, which satisfies,

$$\mathbf{v}_P^* = \mathbf{H}_P\left[\mathbf{v}^*\right] + \widehat{\mathbf{S}}_P^{\mathbf{v}} - \mathbf{D}_P^{\mathbf{v}} \left(\nabla p_P^m\right). \tag{3.44}$$

Therefore, a correction to the velocity field, pressure field, and mass flow rate must ensure mass conservation. Denoting the corrections as $(\mathbf{v}', p', \dot{m}')$, the relations between the exact and the iteratively computed fields are

$$\begin{aligned} p &= p^m + p', \\ \mathbf{v} &= \mathbf{v}^* + \mathbf{v}', \\ \dot{m} &= \dot{m}^* + \dot{m}'. \end{aligned} \tag{3.45}$$

Substituting the mass flux rate given by Eq. (3.45) in Eq. (3.42), the mass conservation equation becomes

$$\sum_{f(P)} \dot{m}'_f = -\sum_{f(P)} \dot{m}^*_f = -\sum_{f(P)} \rho_f \mathbf{v}_f^* \cdot \mathbf{n} \delta S_f, \tag{3.46}$$

where the velocity at the CV-face $\mathbf{v}_f^*$ can be computed again by the Rhie-Chow interpolation as

$$\mathbf{v}_f^* = \overline{\mathbf{v}_f^*} - \overline{\mathbf{D}_f^{\mathbf{v}}} \left(\nabla p_f^m - \overline{\nabla p_f^m}\right). \tag{3.47}$$

In Eq. (3.46), a zero value of the RHS produces a zero correction field that ensures mass conservation. In contrast, a nonzero RHS results in a mass imbalance that violates the conservation statement. Hence, the velocity field $\mathbf{v}_f^*$ involved in causing $\dot{m}_f^*$ must be corrected to enforce conservation, which can be derived by subtracting Eq. (3.44) from Eq. (3.40) for the cells P and F, yielding

$$\begin{aligned} \mathbf{v}'_P &= \mathbf{H}_P\left[\mathbf{v}'\right] - \mathbf{D}_P^{\mathbf{v}}\left(\nabla p'_P\right) \\ \mathbf{v}'_F &= \mathbf{H}_F\left[\mathbf{v}'\right] - \mathbf{D}_F^{\mathbf{v}}\left(\nabla p'_F\right), \end{aligned} \tag{3.48}$$

from which a corrected face velocity is interpolated and expressed as

$$\overline{\mathbf{v}'_f} = \overline{\mathbf{H}_f}\left[\mathbf{v}'\right] - \overline{\mathbf{D}_f^{\mathbf{v}}}\left(\overline{\nabla p'}\right)_f. \tag{3.49}$$

Alternatively, the corrected face velocity derives also by subtracting Eq. (3.47) from Eq. (3.43)

$$\mathbf{v}'_f = \overline{\mathbf{v}'_f} - \overline{\mathbf{D}_f^{\mathbf{v}}}\left(\nabla p'_f - \overline{\nabla p'_f}\right), \tag{3.50}$$

which is used to compute the mass flow rate,

$$\dot{m}'_f = \left(\rho\mathbf{v}' \cdot \mathbf{n}\delta S\right)_f. \tag{3.51}$$

Substituting Eq. (3.51), (3.50), and (3.49) in Eq. (3.46), the pressure-correction equation emerges and is given by

$$\sum_{f(P)} -\rho_f\left(\overline{\mathbf{D}_f^{\mathbf{v}}}\nabla p'_f\right)\cdot\mathbf{n}\delta S_f = -\sum_{f(P)}\rho_f\mathbf{v}_f^*\cdot\mathbf{n}\delta S_f \underline{-\sum_{f(P)}\rho_f\overline{\mathbf{H}_f}\left[\mathbf{v}'\right]\cdot\mathbf{n}\delta S_f}. \tag{3.52}$$

The underlined term is neglected in the original SIMPLE because the correction velocities tend to zero along the iterative process and do not affect the final solution [4, 124, 147].

The left-hand side term of Eq. (3.52) is treated in the FASTEST code as

$$\begin{aligned} \sum_{f(P)} -\rho_f\left(\overline{\mathbf{D}_f^{\mathbf{v}}}\nabla p'_f\right)\cdot\mathbf{n}\delta S_f &= \sum_{f(P)} -\rho_f\overline{\left(\frac{1}{A_P^{\mathbf{v}}}\right)}_f\delta V_f\frac{\left(p'_F - p'_P\right)}{\|\mathbf{x}_F - \mathbf{x}_P\|}\cdot\mathbf{n}\delta S_f \\ &= \sum_{f(P)} -\rho_f\underbrace{\left[\overline{\left(A_P^{\mathbf{v}}\right)}_f^{-1}\cdot\mathbf{n}\delta S_f\right]}_{\tilde{\mathbf{D}}_f^{\mathbf{v}}}\cdot\mathbf{n}\delta S_f\left(p'_F - p'_P\right). \end{aligned} \tag{3.53}$$

The inverse coefficients at the faces $\overline{\left(A_P^{\mathbf{v}}\right)}_f^{-1}$ are interpolated from the nodal values. As a result, the algebraic form of the pressure equation is given by

$$A_P^{p'}p'_P = \sum_{F\sim NB(P)} A_F^{p'}p'_F + S_P^{p'}, \tag{3.54}$$

with the coefficient defined as

$$\begin{aligned} A_F^{p'} &= \rho_f \tilde{\mathbf{D}}_f^{\mathbf{v}} \cdot \mathbf{n} \delta S_f, \\ A_P^{p'} &= \sum_{F(P)} A_F^{p'}, \\ S_P^{p'} &= -\sum_{f(P)} \rho_f \mathbf{v}_f^* \cdot \mathbf{n} \delta S_f. \end{aligned} \tag{3.55}$$

After solving Eq. (3.54) for all central points of the domain with the incomplete LU decomposition method and extrapolate the pressure field to the boundaries, the velocity, mass fluxes, and pressure are corrected with the following equations:

$$\begin{aligned} \mathbf{v}_P^{**} &= \mathbf{v}_P^* - \frac{\left(\Delta_j p'\right)_P \cdot \beta_{ij}^{P,n+1}}{A_P^{\mathbf{v}}}, \\ \dot{m}_f^{**} &= \dot{m}_f^* - A_F^{p'} \left(p'_F - p'_P\right), \\ p_P^* &= p^m + \gamma^p p'. \end{aligned} \tag{3.56}$$

The pressure field is explicitly under-relaxed with a parameter $0 < \gamma^P < 1$, to increase the robustness and convergence rate. For optimum convergence, $\gamma^P = (1 - \gamma^{\mathbf{v}})$ should be set according to reference [162].

Equations (3.36), (3.54), and (3.56) are sequentially solved in each iteration to obtain the new fields of velocity and pressure that satisfy the divergence-free condition. This sequential solution of the equation is known as the **prediction-correction SIMPLE step**.

However, the standard SIMPLE method described above cannot be directly applied for two-fluid flow because the density discontinuity leads to weak coupling between the pressure and velocity fields [47, 143]. Therefore, ***the SIMPLE-VoF method*** described in [182] that fulfills the divergence-free condition for density rates $\rho_1/\rho_2 = 1:10000$ is implemented. This method consists in applying the SIMPLE corrector-equation but removing the density from it. Hence, Eq. (3.55) and Eq. (3.56) become,

$$\begin{aligned} A_F^{p'} &= \tilde{\mathbf{D}}_f^{\mathbf{v}} \cdot \mathbf{n} \delta S_f, \\ A_P^{p'} &= \sum_{F(P)} A_F^{p'}, \\ S_P^{p'} &= -\sum_{f(P)} \mathbf{v}_f^* \cdot \mathbf{n} \delta S_f. \end{aligned} \tag{3.57}$$

$$\begin{aligned} \mathbf{v}_P^{**} &= \mathbf{v}_P^* - \frac{\left(\Delta_j p'\right)_P \cdot \beta_{ij}^{P,n+1}}{A_P^{\mathbf{v}}}, \\ \dot{m}_f^{**} &= \dot{m}_f^* - \rho_f A_F^{p'} \left(p'_F - p'_P\right), \\ p_P^* &= p_P^m + \gamma^p p'_P. \end{aligned} \tag{3.58}$$

Besides, ***two corrections were added in the Rhie-Chow interpolation of the face velocity: the first proposed by Majumdar [132] to eliminate the dependence of the equation on the under-relaxation parameter used for the velocity, and the second one developed by Mencinger [142] to address the abrupt body forces variation*** . Both corrections modify the definition of the face velocity field, Eq. (3.47), as following,

$$\begin{aligned}\mathbf{v}_f^* = & \underbrace{\overline{\mathbf{v}_f^*} - \tilde{\mathbf{D}}_f^{\mathbf{v}}\left(p_F^m - p_P^m\right) + \left[\lambda_F \frac{(\Delta p')_F^j \cdot \beta_{ij}^{F,n+1}}{A_F^{\mathbf{v}}} + (1-\lambda_F)\frac{(\Delta p')_P^j \cdot \beta_{ij}^{P,n+1}}{A_P^{\mathbf{v}}}\right]}_{\text{Rhie-Chow interpolation}} \\ & + \underbrace{(1-\gamma^{\mathbf{v}})\left(\mathbf{v}_f^m - \overline{\mathbf{v}_f^m}\right)}_{\text{Correction due to under-relaxion}} \\ & + \underbrace{\overline{\mathbf{D}_f^{\mathbf{v}}}\left[\left(\lambda_F \mathbf{b}_F^m + (1-\lambda_F)\,\mathbf{b}_P^m\right) - \left(\lambda_F \overline{\mathbf{b}}_F^m + (1-\lambda_F)\,\overline{\mathbf{b}}_P^m\right)\right]}_{\text{Correction due to body forces}}.\end{aligned} \tag{3.59}$$

The overbar denotes face values computed by interpolation.

3.8. Pressure-velocity coupling algorithm

As shown in Section 3.7, the term involving the velocity corrections $\overline{\mathbf{H}_f}\left[\mathbf{v}'\right]$ is neglected because its value is zero at convergence. However, in the initial iterations, this value is significant. Its dismissal results in too large pressure correction values, causing a decrease in the convergence rate even when under-relaxation is used [216]. For this reason, in this research, *extra corrector steps enhance the SIMPLE algorithm* that at each step recover an additional portion of the neglected term, thus improving the convergence rate. The idea follows the PISO algorithm concept [93] and the explicit corrector step proposed in [239].

PISO has two steps. The first is a prediction-correction step identical to the one used by SIMPLE. The second is a corrector step based on the PRIME (Pressure Implicit Momentum Explicit) [133] algorithm. The PRIME step explicitly computes the velocity using the pressure and velocity field resulting from the first step. This new field is then used to implicitly recompute the pressure-corrector equation, ensuring the momentum and continuity equations' fulfillment at the end of each iteration. The second corrector step partially recovers a portion of the neglected term $\overline{\mathbf{H}_f}\left[\mathbf{v}'\right]$ of the first step. So, the pressure correction values are better approximated, which speeds up convergence (see the mathematical demonstration in [147]). Further, under-relaxation is not required since the momentum equation is explicitly solved, which improves stability [147].

Similarly, Yean and Liu [239] considerably improved the SIMPLEC [216] algorithm (an enhanced version of SIMPLE) and the PISO algorithm with an additional explicit corrector step at the corrected velocities. The improvement of the convergence rate was due to the resulting fields satisfy the momentum equation much better. The proposed correction steps can be seen as applying more PRIME steps but with under-relaxation.

Therefore, ***the new pressure-velocity coupling algorithm enables using one or more corrector PRIME steps after the prediction-correction SIMPLE step*** to improve its convergence. In the following, the prediction-correction SIMPLE step will be only called the **SIMPLE step**, and the

correctors PRIME steps will be the **PRIME steps**. Zero PRIME steps retain the standard SIMPLE algorithm. One PRIME step executes the PISO algorithm. Two or more PRIME steps carry out the extra corrections proposed in [239]. This implementation is similar to the idea of the PIMPLE algorithm adopted by the open-source software OpenFOAM [89] to unify the SIMPLE and the PISO algorithms. It is explained in detail in [86, 147].

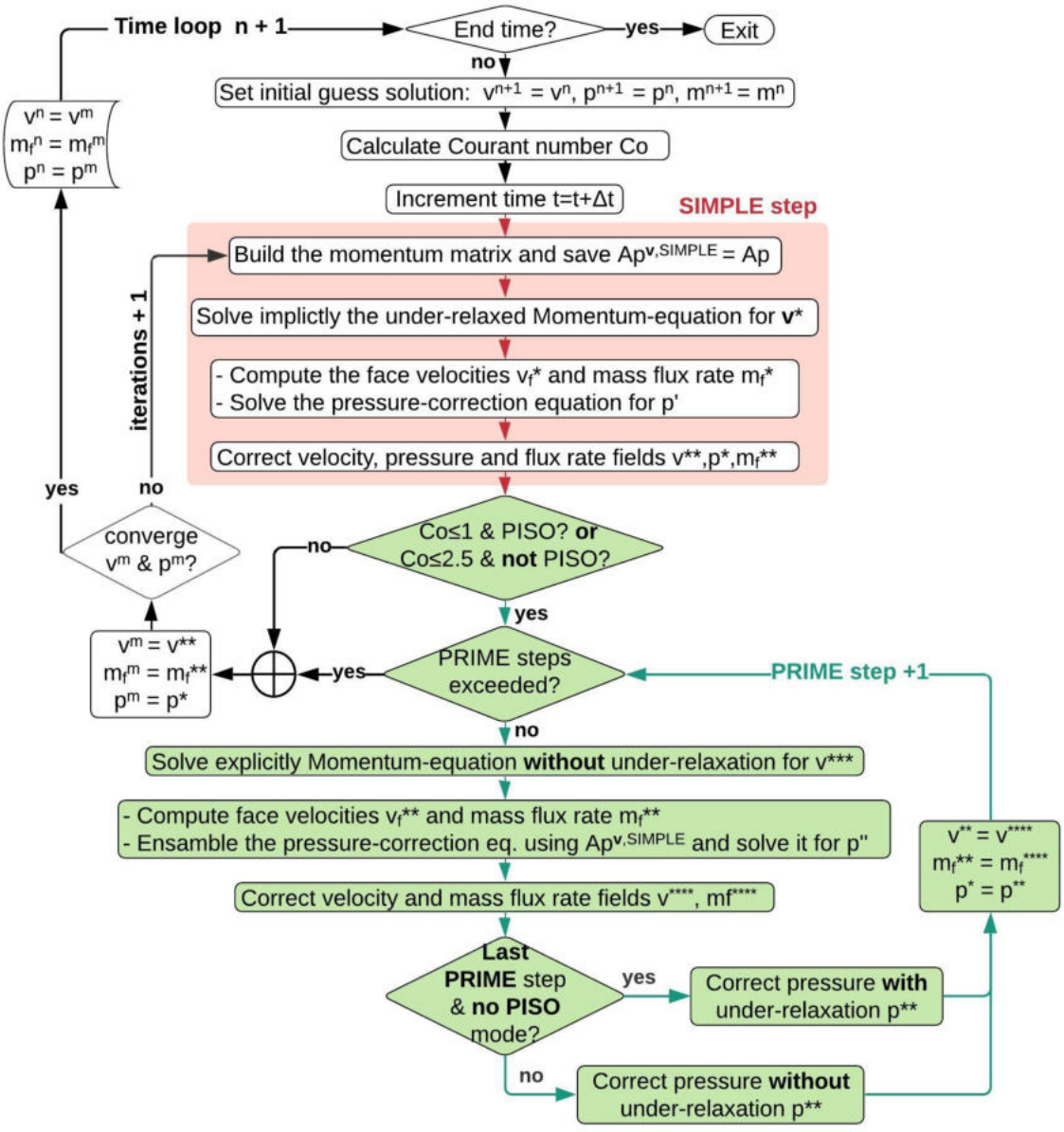


Figure 3.3.: Flow chart of the enhanced SIMPLE algorithm. The green color highlights the new implementations

The flowchart of the enhanced SIMPLE algorithm by PRIME steps for a generic one fluid transient case is shown in Figure 3.3.

At each new time (t_{n+1}), the algorithm performs the following:

1. Set the variables' solution at the last time (t_n) as the initial guess solution at time (t_{n+1}). It is denoted by $\left(\mathbf{v}_P^m, \dot{m}_f^m, p_P^m\right)$ for pressure, velocity, and flow rate fields.

   SIMPLE step:

2. Compute the A-coefficients and the source term for the momentum equation with Eq. (3.35), and save the $A_P^{\mathbf{v}}$ coefficients in a variable $A_P^{\mathbf{v},SIMPLE}$ for later use.

3. Solve the momentum equation (3.36) implicitly to obtain the momentum conservation velocity field $\mathbf{v}^*$.

4. Calculate the velocity at CV-faces $\mathbf{v}_f^*$ with Eq. (3.59) and then update the mass flux rate $\dot{m}_f^*$ with Eq. (3.46).

5. Determine the A-coefficients and the source term for the pressure-correction equation, with Eq. (3.57).

6. Solve the pressure-correction equation (3.54) for the pressure correction field p'.

7. Correct the velocity and pressure fields at the CV-centers and the mass flow rate at the CV-faces applying Eq. (3.58) to obtain the fields that satisfy the mass conservation equation $\left(\mathbf{v}_P^{**}, \dot{m}_f^{**}, p_P^*\right)$.

   PRIME step(s):

8. Recompute the A-coefficients and the source term with Eq. (3.35) for the momentum equation using the latest available velocity, pressure, and mass flow rate fields.

9. Calculate the momentum equation (3.34) explicitly without under-relaxation to obtain a new momentum conservation velocity field $\mathbf{v}^{***}$.

10. Update the face velocity field $\mathbf{v}_f^{***}$ with Eq. (3.59) and the mass flux rate field $\dot{m}_f^{***}$ with Eq. (3.46).

11. Recompute the A-coefficients and the source term for the pressure-correction equation with Eq. (3.57). This time employs the old $A_P^{\mathbf{v},SIMPLE}$ coefficients, and the new face velocities.

12. Recalculate the pressure-correction equation (3.54) for a second pressure corrector field p''.

13. Update pressure, velocity, and mass flow rate fields to get $\left(\mathbf{v}_P^{****}, \dot{m}_f^{****}, p_P^{**}\right)$ with Eq. (3.58), but using instead of $A_P^{\mathbf{v},SIMPLE}$, and not under-relax the pressure unless it is the last corrective step and it is not working in PISO mode.

14. Go to step **8** and repeat depending on the number of correction steps desired, setting $\left(\mathbf{v}_P^{**}, \dot{m}_f^{**}, p_P^*\right) = \left(\mathbf{v}_P^*, \dot{m}_f^{****}, p_P^{**}\right)$.

15. Go back to step **2**, set $\left(\mathbf{v}_P^m, \dot{m}_f^m, p_P^m\right) = \left(\mathbf{v}_P^{****}, \dot{m}_f^{****}, p_P^{**}\right)$ and repeat until convergence.

The solution is assumed to have converged when the normalized residual of each dependent variable $\varphi = \mathbf{v}, p$ defined by Eq. (3.60) is less than the chosen convergence criterion ε.

$$R(\varphi) = \frac{\max\limits_{\text{all cells}} \left| A_P \varphi_P - \sum\limits_{F(P)} A_F \varphi_F - S_P^{\varphi} \right|}{\max\limits_{\text{all cells}} |A_P \varphi_P|} \leq \varepsilon \tag{3.60}$$

For steady-state and transient-state simulations with high Courant (Co) numbers, the SIMPLE algorithm is automatically set as default. Whereas, for transient simulations with Co less than 2.5, the new algorithm with extra internal corrections can be used, and the PISO Mode (one PRIME step) is active only for Co less than one. These conditions ensure stability and result from several tests, such as the one presented below in Section 3.8.2.

3.8.1. Code verification of the pressure-velocity coupling algorithm

The code verification process demonstrates that a partial differential equation code correctly solves its governing equations [27]. Here, the proposed pressure-velocity coupling algorithm is verified using the method of manufactured solutions (MMS). Salari and Knupp [106] provides a comprehensive guide to creating manufactured solutions and the procedure for inserting them into the code and analyzing the results. The MMS checks whether the code's numerical algorithm reaches its theoretical order of accuracy as the domain is refined [174]. In this case, the FASTEST code uses the numerical methods introduced in Sections 3.5 and 3.7, which are theoretically second-order accurate. Consequently, this is the order which needs to be observed to verify the new implementations.

For the verification, simplifications were made by assuming a single 3D incompressible fluid in a fixed domain to avoid regions with abrupt variations of physical properties that may affect the order-of-accuracy verification as recommended in [106]. Thus, the governing equations for this problem become

$$\nabla \cdot \mathbf{v} = 0, \tag{3.61}$$

$$\rho \frac{\partial \mathbf{v}}{\partial t} + \rho \nabla \cdot (\mathbf{v}\mathbf{v}) = -\nabla p + \nabla \cdot \mu \left[\nabla \mathbf{v} + (\nabla \mathbf{v})^T \right] + \mathbf{S}^{\mathbf{v},MMS}. \tag{3.62}$$

Let the vector $\mathbf{S}^{\mathbf{v},MMS}$ be a source term inserted into the momentum equation strictly for reproducing the manufactured solutions. $\mathbf{S}^{\mathbf{v},MMS}$ outcomes from substituting the manufactured solutions into the momentum equation and subsequent derivation of it.

The manufactured solutions for the dependent variables are functions of the form

$$\begin{aligned} u(\mathbf{x},t) &= 0.5 + 0.5\cos(\pi x)\cos(\pi y)\cos(\pi z)\cos(\pi \mathrm{t}), \\ v(\mathbf{x},t) &= 0.5 + 0.25\sin(\pi x)\sin(\pi y)\cos(\pi z)\cos(\pi \mathrm{t}), \\ w(\mathbf{x},t) &= 0.5 + 0.25\sin(\pi x)\cos(\pi y)\sin(\pi z)\cos(\pi \mathrm{t}), \\ p(\mathbf{x},t) &= 0.5 + \cos(\pi x)\cos(\pi y)\cos(\pi z)\cos(\pi \mathrm{t}). \end{aligned} \tag{3.63}$$

These functions are smooth and satisfy the divergence-free condition required for incompressible fluids, as is suggested in [8, 23, 27, 221].

The coverage test suite's computational domain is a cube of side 1 m filled with a fluid of density $\rho = 1$ kg/m^3 and the dynamic viscosity $\mu = 1$ Pa·s. All the boundaries are defined as inlet boundary conditions. Thus, the values of the velocity components are computed from the manufactured solutions and prescribed there. In contrast, the manufactured pressure solution is only given for a point in the domain as a pressure reference point. Further, because the test is unsteady, the manufactured solutions evaluated at the initial time are used as the initial input condition to assure that the solution at any time will tend to zero when the mesh is refined [181].

Five orthogonal and uniform grids with a refinement ratio of $r = 2$ are used for the grid convergence testing. The grids consist of 8^3, 16^3, 32^3, 64^3, and 128^3 hexahedral control volumes. The convective term is approximated with the CDS and the temporal term with the BDF2 scheme. Each run finishes after 100 time-steps of constant $\Delta t = 10^{-4}$ s. So, the temporal discretization error keeps constant concerning grid refinement. The enhanced SIMPLE algorithm is set to make two PRIME steps after each SIMPLE step. The convergence tolerance is the normalized residuum of 10^{-12} for all variables, and the maximum number of iterations per time-step is limited to 100. The under-relaxation factors are 0.7 for the velocity field and 0.3 for the pressure field.

After 100 time-steps, the normalized global discretization error of each dependent variable at each grid level is defined by

$$e_2^{\phi(grid)} = \sqrt{\frac{\sum_n \left(\phi_n^{mms} - \phi_n^d\right)^2 \delta V_n}{\sum_n \delta V_n}}, \tag{3.64}$$

where n denotes the index of the discrete solution location, and δV_n is the local volume. ϕ_n^{mms} denotes the exact solution of a dependent variable evaluated at position $\mathbf{x}_n$ and time t, while ϕ_n^d is its discrete solution at the same point in space and time. The difference between the discrete and exact solution is the local discretization error.

The observed order-of-accuracy denoted by h of each variable is estimated from the expression

$$h_{(grid_2)}^{\phi} = \frac{\ln\left[\dfrac{e_2^{\phi(grid_1)}}{e_2^{\phi(grid_2)}}\right]}{\ln(r)}. \tag{3.65}$$

Figure 3.4 shows the local discretization errors for the x-velocity u, y-velocity v, z-velocity w, and pressure p for the finest mesh at 100 time-steps. As expected at this mesh refinement level, the solutions are very close to the manufactured solutions. The local errors are relatively small, with maximum values of about 3.5×10^{-5} m/s for the velocity field and 3.5×10^{-2} Pa for the pressure field. The higher inaccuracies are located on the boundaries.

Table 3.1 lists the normalized global discretization errors at each grid level of the three velocity components and the pressure field. For better visualization of the error decay order through mesh refinement, the computed e_2 quantities of each variable and the theoretical second-order are plotted in Figure 3.5 using a logarithmic scale. N symbolizes the number of control volumes of each grid level.

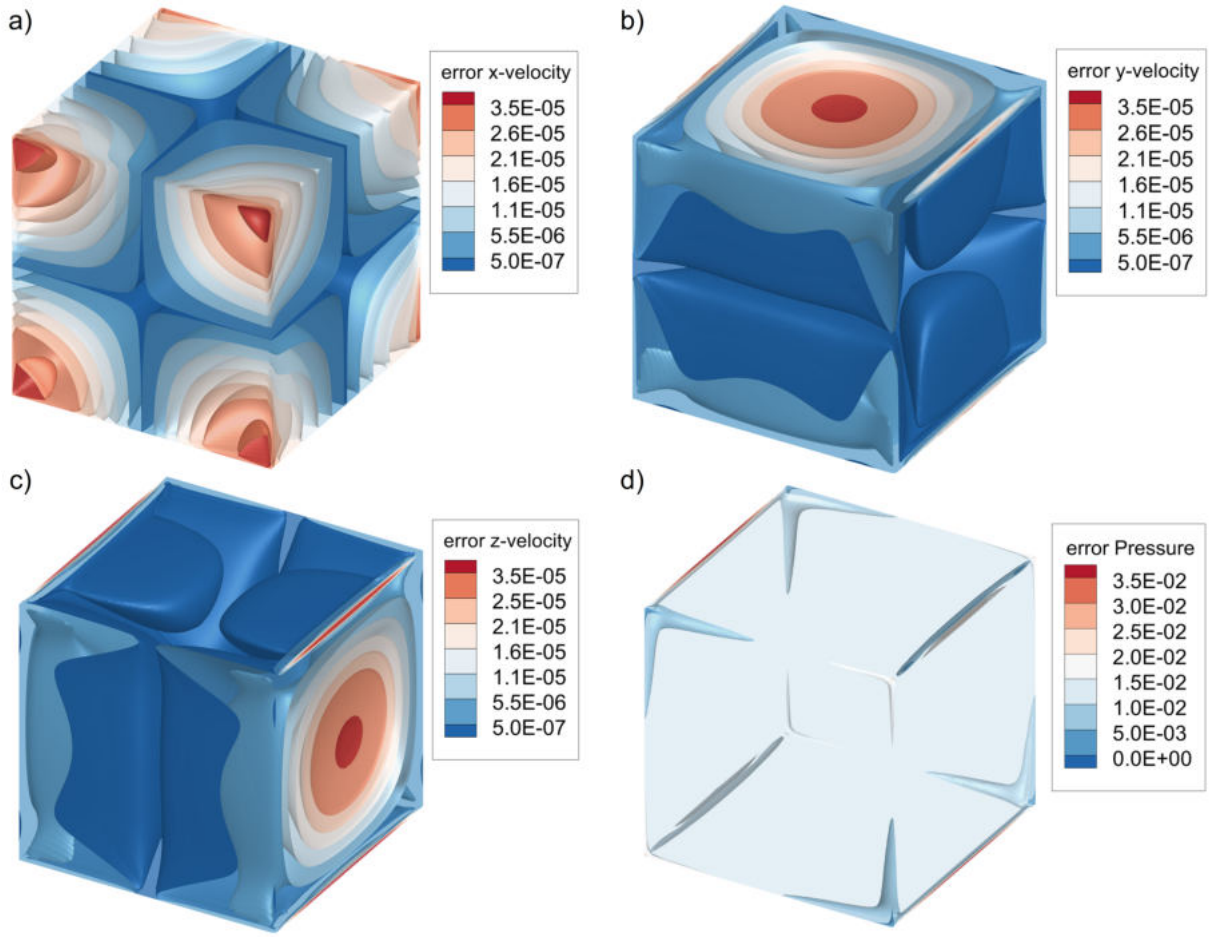


Figure 3.4.: Iso-surfaces at contours group levels of the local discretization errors for the velocity on the x, y, and z-directions and the pressure field at 100 time-steps using the finest grid

Grid	N	u		v		w		p	
		e_2	h	e_2	h	e_2	h	e_2	h
1	8^3	2.92E-03	—	1.64E-03	—	1.70E-03	—	3.83E-02	—
2	16^3	7.53E-04	1.96	4.41E-04	1.90	4.46E-04	1.94	1.34E-02	1.51
3	32^3	1.89E-04	1.99	1.13E-04	1.96	1.14E-04	1.97	4.41E-03	1.60
4	64^3	4.75E-05	2.00	2.85E-05	1.99	2.86E-05	1.99	1.39E-03	1.67
5	128^3	1.19E-05	2.00	7.14E-06	2.00	7.16E-06	2.00	4.11E-04	1.76

Table 3.1.: Grid refinement results for the proposed pressure-velocity coupling algorithm simulating a single incompressible 3D fluid.

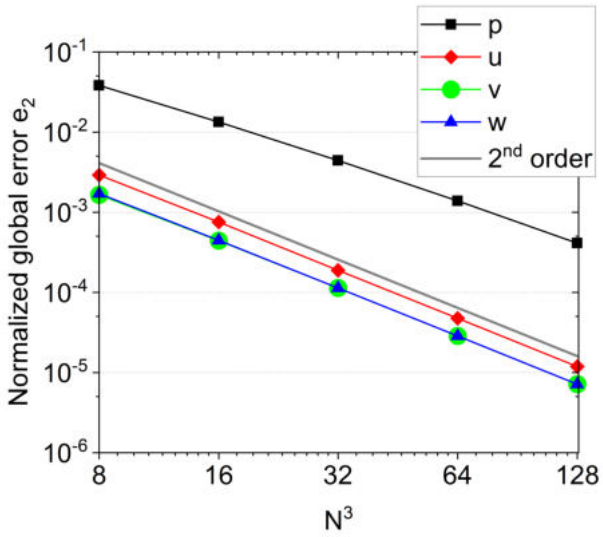


Figure 3.5.: Decay lines of the normalized global error e2 of the three velocity components and the pressure and the theoretical second-order decay

Conclusions:

The test suite results show second-order accuracy behavior for the three velocity components and a continuous decrease in the pressure field's normalized global error. It achieves an order of 1.76 for the most refined grid. Therefore, the convergence test results confirm that the new pressure-velocity coupling algorithm is free of code mistakes.

3.8.2. Validation and performance analysis of the new pressure-velocity coupling

The new pressure-velocity coupling's correctness is validated solving the Lid-driven Cavity problem with Reynolds number of 1000. It consists of solving the velocity and pressure field of a fluid of density $\rho = 1$ kg/m^3 in a 2D square cavity of 1 m sides. The cavity's top wall moves with a constant horizontal velocity equal to 1 m/s.

The computational domain consists of 120 $\times$ 120 uniform CVs, and the four walls are set as no-slip boundary conditions. The unsteady Navier-Stokes equations are solved several times until reaching the steady-state. The convective term is approximated with the QUICK scheme and the time term with the BDF time scheme using $\Delta t = 10^{-2}$ s, which yields a maximum $Co = 1.13$. No under-relaxation for the velocity and pressure is used, and the convergence criterion for the three velocity components and pressure field is 10^{-6}. The pressure-coupling algorithm performs a maximum of 100 iterations composed of a SIMPLE step and two PRIME steps per time-step.

Figure 3.6 shows a) the streamlines and b) the vortices contours in the square cavity at steady state ($t = 90$ s). While Figure 3.7 contrasts the resulting a) horizontal velocity u across the vertical center plane and b) vertical velocity v across the horizontal center plane with the reference result taken from [238]. The velocity results show a reasonably good agreement with the benchmark available data. The root-mean-square (RMS) error between the numerical and reference results for the horizontal velocity u is 0.0017, and for the vertical velocity, v is 0.0041.

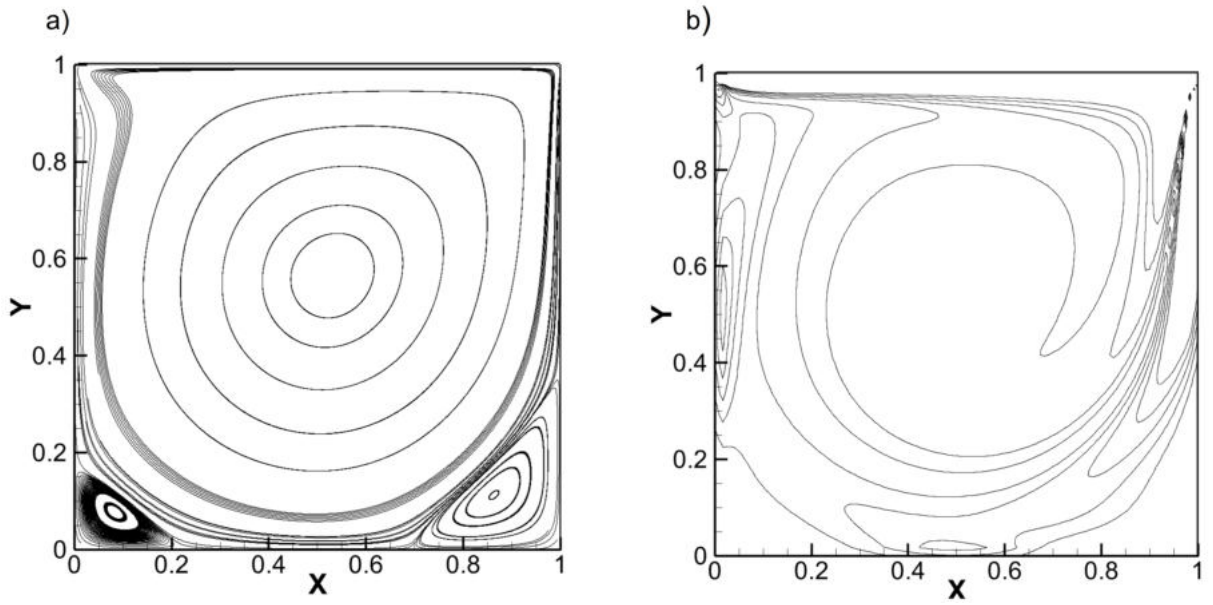


Figure 3.6.: a) Streamlines, manual levels and b) vortices contours, level from -5 to 5 with step 1 of the Lid-driven Cavity problem with Re=1000 at steady state ($t = 90$ s)

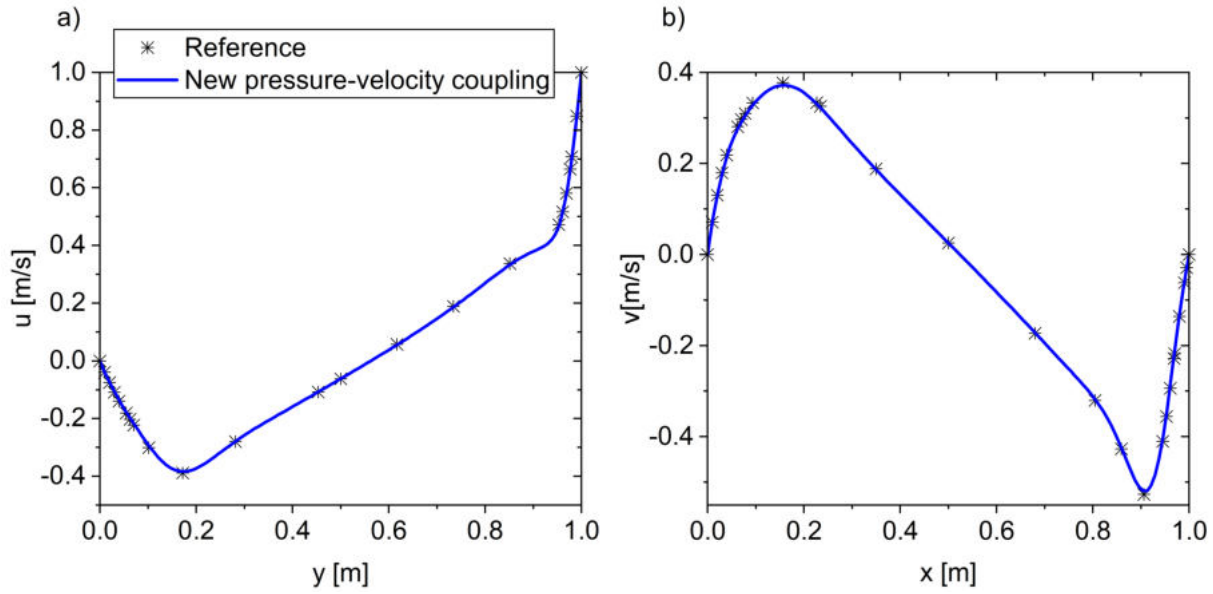


Figure 3.7.: Comparison of the results for the a) horizontal velocity component (u) across the vertical plane $x = 0.5$ m and b) vertical velocity component (v) across the horizontal plane $y = 0.5$ m, with the Reference data published in [238]

A second aim with this test case is observing the influence of the number of PRIME steps on the solver performance. A performance index is the average computational time that requires the

solution algorithm to solve a problem correctly. It is strongly linked to the number of iterations (SIMPLE step + PRIME steps) performed during the simulation to achieve the solution. Table 3.2 collects the total number of iterations that perform the solution algorithm when zero to five PRIME steps are setting and the total real-time each one consumes. Note that zero PRIME steps refer to the SIMPLE algorithm, and one executes the new algorithm in PISO mode. The computations were performed without and with under-relaxation since for the SIMPLE mode, it is an essential factor for the success of the solution, and the PISO mode requires no relaxation. For this analysis, only the first 30 s are computed since the significant changes of the velocity and pressure fields finish, and the steady regimen appears after this period.

PRIME steps	**No under-relaxation**		$(\gamma^{\mathbf{v}}, \gamma^{p}) = (0.8, 0.2)$		$(\gamma^{\mathbf{v}}, \gamma^{p}) = (0.6, 0.4)$	
	Iterations	Time [s]	Iterations	Time [s]	Iterations	Time [s]
0(SIMPLE)	*	—	26841	957	27494	980
1(PISO)	11080	628	12135	688	14265	809
2	8037	652	9545	774	9511	772
3	7341	762	8336	865	8327	864
4	6939	738	7490	796	7556	803
5	6677	784	6846	803	*	—

Table 3.2.: Total number of iterations and total real-time resulting from setting different numbers of PRIME steps for the Lid-driven Cavity problem with $Re = 1000$ and a maximum Co=1.13, * denotes divergence

The first correction (PISO) can decrease the total number of iterations by about 50% of those required by the SIMPLE algorithm. In real-time, it represents from 20% to 30% less computational time. The second correction reduces almost 65%, the third 70%, the fourth 72%, and the fifth 74% of the iterations compared to SIMPLE. However, from the second correction, the reduction of consuming time is less significant, with two PRIME steps being almost 20%, with three 15%, with four 4%, and with five 3% compared to the time consumed by SIMPLE. Figure 3.8 shows the numerical normalized residuals for velocity and pressure fields using 0-3 PRIME steps.

PRIME steps	**max** $Co = 0.56$		**max** $Co = 2.82$		**max** $Co = 5.62$	
	Iterations	Time [s]	Iterations	Time [s]	Iterations	Time [s]
0 (SIMPLE)	44621	1760	16096	574	11816	421
1 (PISO)	20995	1400	23183	1315	566047	32098
2	16134	1360	5744	466	9828	797
3	13751	1210	22711	2357	*	*

Table 3.3.: Total number of iterations and computational time for solving the Lid-driven Cavity problem at different maximum Co numbers and using under-relaxation factors of $(\gamma^{\mathbf{v}}, \gamma^{p}) = (0.6, 0.4)$. * denotes divergence

In summary, for a maximum $Co = 1.13$, the computation using the PISO mode without under-relaxation requires the least time. In contrast, the computations with the SIMPLE mode are the slowest and without under-relaxation diverge, showing the importance of using relaxation factors for an adequate functionality of SIMPLE. Furthermore, at this Co condition, a maximum of thee

PRIME steps are the best option to improve the convergence rate and consume a reasonable computation time.

To finish this analysis, Table 3.3 contrasts the behavior of the solution algorithm for solving this test case at different Co numbers using under-relaxation factors $(\gamma^{\mathbf{v}}, \gamma^{p}) = (0.6, 0.4)$. For $Co = 0.56$, three PRIME steps is the fastest option, this requires 30% of the iterations done by SIMPLE and reduce about 30% of the computational time. For $Co = 2.82$ and $Co = 5.62$, the PISO mode requires more iterations and time than the other options. For these cases, two PRIME steps seem to be a good option, although for the large Co, SIMPLE is the best alternative.

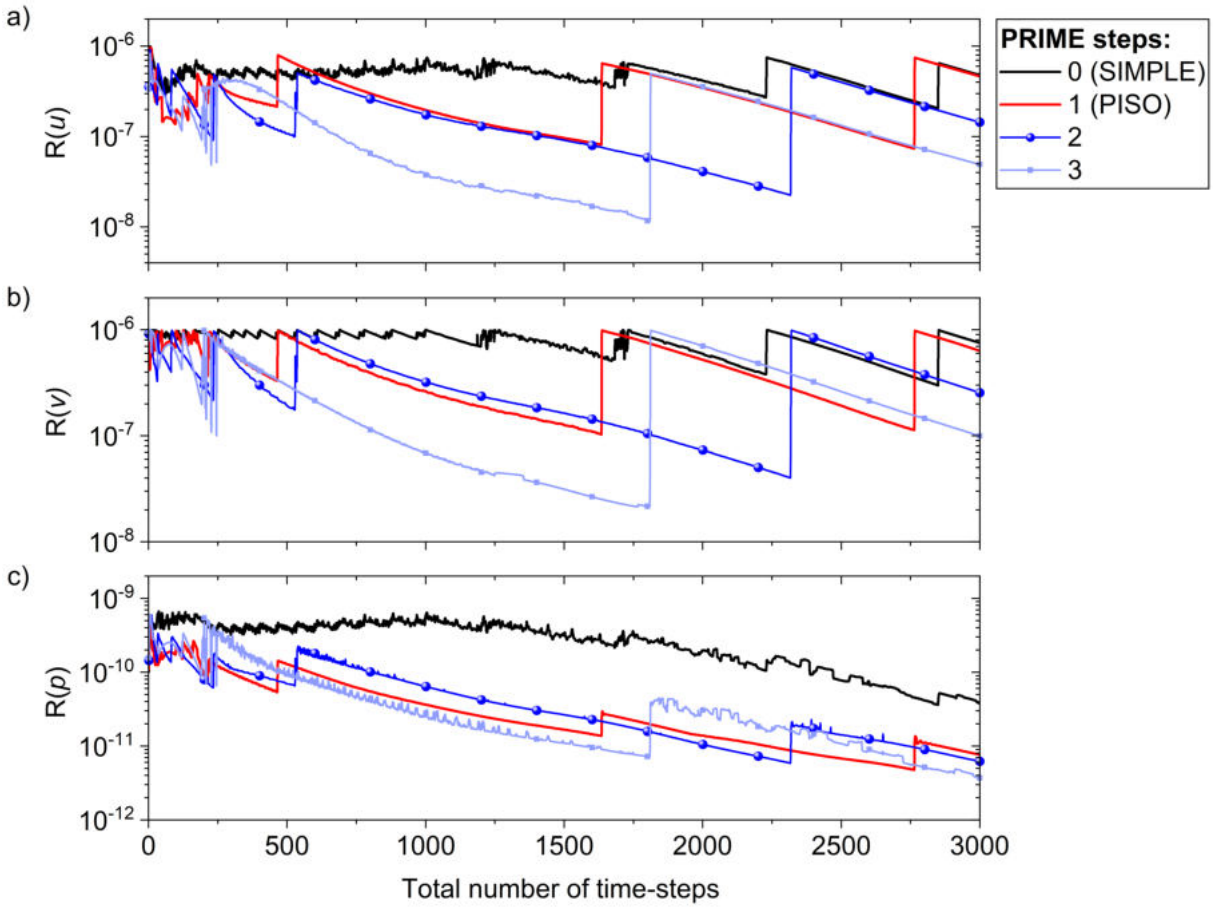


Figure 3.8.: Normalized residuals of velocity components a) u, b) v, and c) the pressure field resulting from 0-3 PRIME steps when the $Co = 1.13$ and the under-relaxation factors $(\gamma^{\mathbf{v}}, \gamma^{p}) = (0.8, 0.2)$ are applied

Conclusions:

- The pressure-velocity coupling algorithm has been successfully validated. The resulting velocities for the Lid-drive Cavity problem with Re=1000 agree with the reference results, and the streamlines and vortices contours coincide with the physics of the problem.

- The PRIME steps improve convergence rate and numerical accuracy, which reduces the total number of iterations. However, they do not directly implicate a reduction in the computational effort because each extra PRIME step adds a resolution of the momentum and pressure-correction equations. For $Co \leq 1$, one PRIME step (PISO mode) shows notable positive effects in convergence rate and decrease of computational effort. For moderate Co, two or three PRIME steps also improves efficiency, while for large Co, zero PRIME steps (SIMPLE mode) are the best option. The ideal number of extra correction PRIME steps depends on the case. One needs to be careful because too many internal correction steps can lead to divergence of the solution as the momentum equation is solved explicitly, especially for large Courant numbers.

3.8.3. Role of the corrections for body forces and under-relaxation in the Rhie-Chow interpolation

Correcting the Rhie-Chow interpolation dependence on the under-relaxation factors and managing abrupt body force changes are essential tasks to approximate the face velocities in collocated formulations adequately. The importance of the correction of Majumdar [132] to make the Rhie-Chow interpolation independent of the velocity under-relaxation factors have been discussed in [102, 156, 243]. Some applications of it can be reviewed in [98, 124, 211]. Similarly, references [47, 143] emphasize that the proper calculation of the gravity force in the presence of density discontinuities reduces the apparition of the undesired spurious currents near the flow interface. Therefore, the two corrections were implemented in the FASTEST code. In this section, their influence in the global solution is investigated using the one-dimensional case presented in [142]. Two fluids under the effect of gravity are kept at rest by setting the velocity at the boundary to zero. Thus, the pressure gradient must counterbalance the body forces field.

The 1D domain is placed in $0 < x < 0.1$ m and contains two static fluids, air and water, distributed according to the volume fraction

$$\alpha(x) = \begin{cases} 1 & 0.030 \text{ m} < x < 0.062 \text{ m} \\ 0 & otherwise \end{cases} \tag{3.66}$$

The variable α is one for air and zero for water. The material properties of air are $\rho_1 = 1.29$ kg/m^3 and $\mu_1 = 1.8 \times 10^{-5}$ Pa·s, and for water $\rho_2 = 1000$ kg/m^3 and $\mu_2 = 1.0 \times 10^{-3}$ Pa·s. The body force field is given by $b(x) = g\left[\alpha\rho_1 + (1-\alpha)\rho_2\right]$ where $g = 10$ m/s^2.

The domain is discretized with three uniform grids containing 20, 40, and 80 cells. Though the problem addresses two fluids at steady rest, the calculation is treated as transient case and run until they achieve and keep in the steady-state, 100 time-steps equal to 0.001 s. The used time scheme is the BDF2, the convergence criterium for the variables is 10^{-12}, and the pressure-velocity coupling is set to PISO mode.

For first testing, only the correction due to body forces, without under-relaxation, is applied. The three grid levels are computed with a) the original Rhie-Chow interpolation and b) the Rhie-Chow interpolation, including the correction for body forces. Option b) also involves the redistribution of the body forces in the momentum equation. Figure 3.9 shows the pressure and velocity fields along the x-domain for the two options. The pressure fields resulting from options a) and b) follow a linear growth tendency in the water region and constant pressure in the air region, as

expected. However, the results from the a) Rhie-Chow interpolation exhibit a zigzagging pattern that becomes less evident as the mesh refines. Although, in this case, it does not overwhelm the solution, it can be a difficulty in dynamic multifluid flows[144].

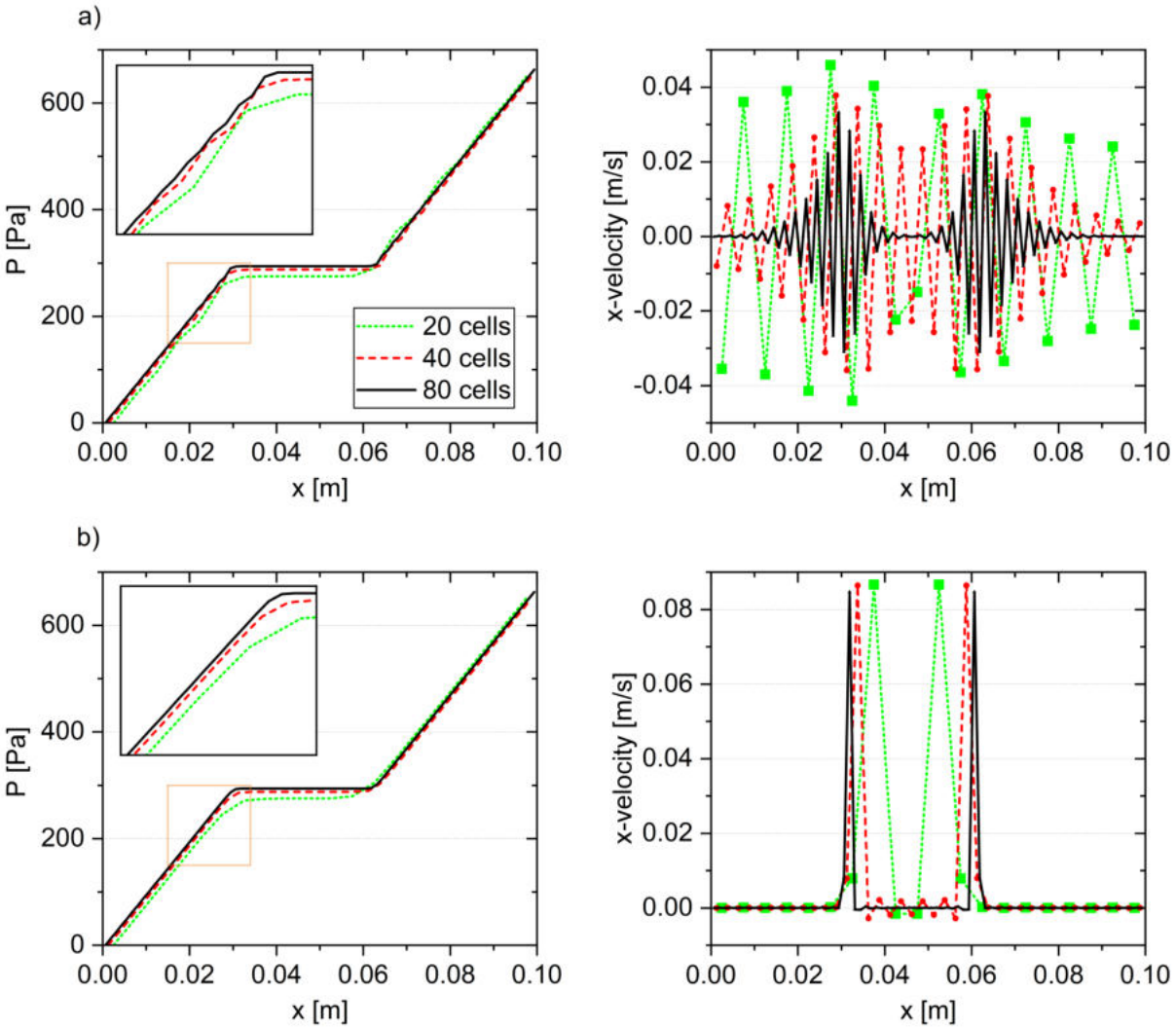


Figure 3.9.: Velocity field along the x-domain at 100 time-steps for 20, 40, and 80 cells computed with a) the original Rhie-Chow interpolation and b) the Rhie-Chow interpolation including the correction for body forces

On the other hand, the velocity field computed with options a) and b) present unphysical velocities since the expected velocity is zero in the whole domain. Especially option a) has a high level of parasitic currents. The maximum amplitude of these false velocities is around 0.9 m/s near the discontinuities and slightly decreases with the mesh's refinement. Using option b), the velocity field is almost zero in the whole domain except in the lighter fluid zone. The little zigzag pattern disappears with grid refinement. Though, two unphysical peaks of amplitude 0.9 m/s at the beginning and end of the air region remain.

The second part of this investigation focuses on observing the effect of using the correction to eliminate the dependence on velocity under-relaxation factors in the definition of the face velocities. For this purpose, the one-dimensional test discretized with 80 cells is computed again, but this time using relaxation factors of $0.3 \leq \gamma^{\mathbf{v}} \leq 1.0$ for the velocity, and for the pressure, $\gamma^{p} = (1 - \gamma^{\mathbf{v}})$.

Figure 3.10 presents the resulting velocity field calculated with the Rhie-Chow interpolation a) without and b) with correction for dependence on velocity under-relaxation factors. For the two scenarios in almost the whole domain, the relaxation factors do not influence the resulting velocity field. However, near the discontinuities, the no use of the correction for under-relaxation factors, option a), induces significant differences in the resulting values. Thus, as more under-relaxation is applied, the resulting velocity is further away for the reference values assuming the values without relaxation are the reference. The reduction of the velocity value is proportional to the relaxation factor, e.g., the velocity computed with $\gamma^{\mathbf{v}} = 0.3$ is 30% of the velocity computed with $\gamma^{\mathbf{v}} = 1.0$. In contrast, using option b) interpolation with correction, the resulting velocity field remains the same, regardless of the under-relaxation factor applied.

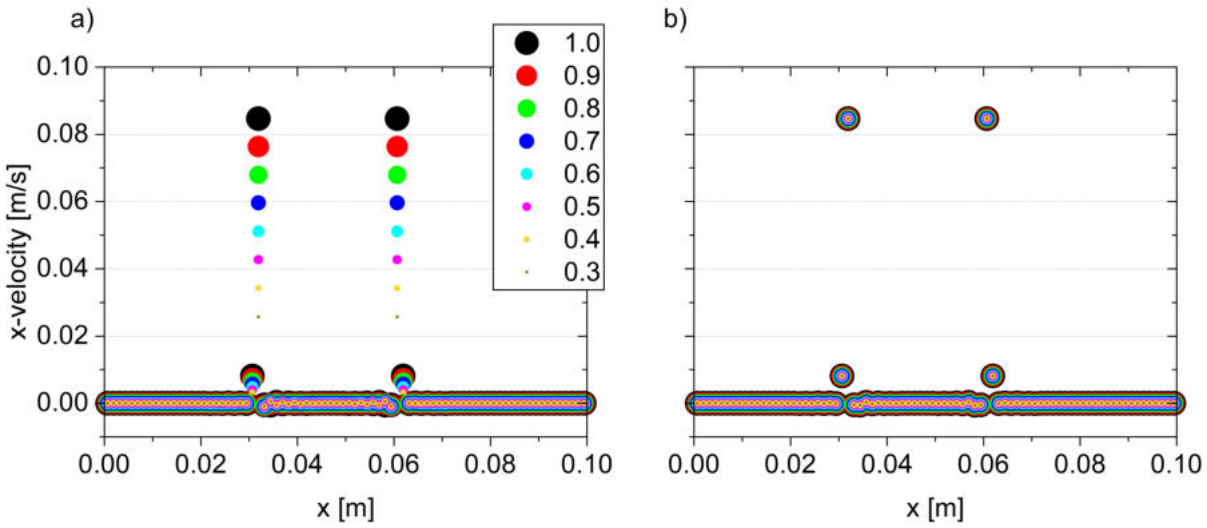


Figure 3.10.: Velocity field along the x-domain at 100 time-steps for 80 cells computed using the Rhie-Chow interpolation a) without and b) with correction for dependence on velocity under-relaxation factors

Conclusions:

- As is shown by this simple test case, the complete elimination of the parasitic currents is practically unfeasible. However, the Mencinger correction can be considered a reasonably good approximation with less parasitic currents than the original Rhie-Chow interpolation.
- According to the last analysis, the correction of Majumdar is essential to ensure that the two-fluid flow problems obtain a velocity field independent of the relaxation factors.

4. The discretization of the VoF equation and the two-fluid flow solver

As described in Chapter 2, the Volume of Fluid (VoF) method completes the one-fluid formulation of multiphase flows. The volume fraction of one of the fluids acts as an indicator function that tracks the instantaneous distribution of the fluids. As the fluids move and the location of the fluid-interface changes, the volume fraction must be updated.

Updating the indicator function is critical for the simulation of multiphase flows. However, this is not a trivial job due to the purely convective nature of the volume fraction transport equation [118, 203]. The VoF equation is susceptible to false diffusion that can smear the flow interface. The numerical solution demands a sophisticated advection scheme that should neither produce numerical diffusion nor unbounded values [149]. Thus, preventing oscillatory behavior in the solution and keeping a sharp interface [147, 203].

One other aspect to keep in mind is that the VoF method is designed to capture sharp free surfaces between immiscible fluids. The free surface must not move across more than one CV in a time-step to capture the interface correctly. Otherwise, the solver will smear the interface over multiple CVs, causing a numerical mixture. Although this mixture could appear as a real mixture, and in some cases, represent a physical mixture, the modeling of the interaction between phases does not exist. Thus, once a numerical mixture occurs, it will not separate into distinct phases again and will continue to damage the results, particularly in closed domains. Therefore, the Co number needs to be kept in a range in which the advection scheme can define a sharp flow interface.

In addition, when the advection schemes are used to capture free surface flows in interaction with flexible structures (FSI applications), the schemes' numerical stability at high Courant numbers and the computation efficiency are also essential requirements [144]. The frequent deformation of the control volumes inside the FSI cycle often induces Co larger than one, and the FSI implicit iterative process alone is very time-consuming.

We can outline that keeping a sharp flow interface requires, on the one hand, a compressive advection scheme, ideally. However, on the other hand, because the solution is discontinuous, the boundedness of the diffuse scheme is needed as well [32]. Hence, commonly a mixture between them is desirable. Likewise, the scheme's accuracy ought to be good even with high Courant numbers ($Co \approx 1$), to allow the use of reasonable time-steps [159]. Thus, reducing the high computational costs usually associated with multiphase flow computations, which is the bottleneck in practical industrial applications.

This chapter begins with the discretization process of the volume fraction transport equation in Section 4.1, followed by an overview of the high-resolution advection schemes in Section 4.2. Whereas Section 4.3 gives a general definition of the interface capturing schemes. The mathematical basis of the schemes lies in the Normalized Variable Formulation (NVF) [117, 119]. Due to

the interface capturing schemes' composed nature, they can not be directly implemented in the VoF equation and require a particular numerical method to include them in the algebraic solution equation. The advantages and disadvantages of the following methods: Deferred Correction (DC) [177], Downwind Weighting Factor (DWF) [120], Normalized Weighting Factor (NWF) [34], and Reviewed Normalized Weighting Factor (RNWF) [28] are pointed out in Section 4.4. Whereas in Section 4.5, the new, more efficient methodology called Modified Normalized Weighting Factor (MNWF) [135] developed in this research's scope is described in detail. The chapter continues with implementing the boundary conditions for the VoF equation in Section 4.6 and the control of the Courant number to avoid excessive smearing of the flow interface in Section . The theoretical part concludes in Section 4.8, where the solution algorithm for multifluid flows in fixed domains is presented.

Section 4.9 presents the solution of two well-known benchmark multifluid cases using six interface capturing schemes: CICSAM, MCICSAM-W, MCICSAM-Z, HRIC [149], FBICS [204], and CUIBS [159] numerically implemented with the new MNWF method in order to find suitable advection schemes for working at moderate-high Courant numbers that can be used later for moving domains. Finally, in Section 4.10, the more efficient and accurate interface capturing schemes resulting from the previous part are used to solve more challenging problems and validate the complete two-fluid flow solver.

4.1. Discretization of the volume-fraction transport equation

Likewise, the momentum equation, the volume fraction transport equation (2.9), is discretized in space with the FVM method and in time with the FD method. The VoF equation is integrated over each control volume, and the volume integral of the divergence of α is transformed to a surface integral applying the Gauss theorem. Hence, the integral form of the volume fraction transport equation for a control volume P in moving domain results in:

$$\underbrace{\frac{\partial}{\partial t}\int_{V_P} \alpha dV}_{\text{transient term}} + \underbrace{\int_{S_p} (\mathbf{v}-\mathbf{v}_g)\,\alpha\cdot\mathbf{n}dS}_{\text{convection term}} = 0. \tag{4.1}$$

The transient term is discretized with one of the implicit time schemes already presented in Section 3.4.5. Implicit Euler or BDF2, with a constant or adaptative time-step. Whereas the mid-point ruler numerically approximates the surface integral.

Using the BDF2 time scheme with an adaptative time-step, Eq. (4.1) becomes

$$k_P^{n+1}(\alpha_P\delta V_P)^{n+1} + k_P^{n}(\alpha_P\delta V_P)^{n} + k_P^{n-1}(\alpha_P\delta V_P)^{n-1} = \\ -\sum_{f(P)}\left[\mathbf{v}_f^n\cdot\mathbf{n}\delta S_f^{n+1} - \mathbf{v}_{g,f}^{n+1}\cdot\mathbf{n}\delta S_f^{n+1}\right]\alpha_f^{n+1} = -\sum_{f(P)} F_f^{\alpha}\alpha_f^{n+1}, \tag{4.2}$$

where Eq. (3.33) defines the coefficients k and the superscripts $n+1$, n, and $n-1$, denote a variable value at time t_{n+1}, t_n, and t_{n-1}, respectively.

Observe that the face flow velocity $\mathbf{v}_f$ is the value at the last time t_n as for a small enough

time-step, the velocity variation is negligible compared to the considerable variation of α, thus using the last updated values is reasonable [209]. The difference between the flow velocity and the grid velocity multiplied by the face surface yields the relative volumetric flux at CV-face f denoted by F_f^α. Obviously, for a fixed domain, $\mathbf{v}_{g,f} = 0$, and the volume and face surfaces remain constant in time.

After some arithmetic operation, the algebraic form of the discretized equation reads

$$\left(A^\alpha_{P(time)} + A^\alpha_P\right)\alpha_P = \sum_{F(P)} A^\alpha_F \alpha_F + S^\alpha_P, \tag{4.3}$$

with

$$\begin{aligned} A^\alpha_{P(time)} &= k_P^{n+1}\delta V_P^{n+1}, \\ S^\alpha_{P(time)} &= -\left[k_P^n(\alpha_P\delta V_P)^n + k_P^{n-1}(\alpha_P\delta V_P)^{n-1}\right], \\ S^\alpha_P &= S^\alpha_{P(time)}, \end{aligned} \tag{4.4}$$

and the coefficient A^α_P and A^α_F are defined according to the advection scheme used to approximate the face volume fraction α_f^{n+1}.

It is worth mentioning that the volume of fluid transport equation is a purely convective equation, so the numerical stability of the solution at large time-steps highly relies on the grade of the independence of the advection scheme from the local Courant number [83, 159] which is defined at each CV-face by

$$Co_f = \frac{F_f^\alpha \cdot \Delta t}{\delta V_P}, \tag{4.5}$$

and for the CV-center as

$$Co_P = \sum_{f(P),out} |Co_f|. \tag{4.6}$$

The local Co of the control volume P is the sum of the Courant number of each outflow face of the control volume [209] and represents the worst-case condition.

4.2. High-resolution advection schemes

A high-resolution (HR) advection scheme is a composite high-order scheme combined with the Convective Boundedness Criterion (CBC) of Gaskell and Lau [66] to ensures that the interpolation profile at the cell face does not underflow or overflow the cell [147]. Some examples are SUPERBEE [176], MUSCL [217], SMART, or STOIC. The HR schemes can be formulated in the framework of the Normalized Variable Formulation (NVF) procedure and plotted for analysis in the Normalized Variable Diagram (NVD) introduced by Leonard in [117, 119].

The NVF is a face formulation technique based on the local normalization of the variables involved in approximating the value α_f at face f [147]. The technique utilizes the donor α_D, acceptor α_A, and upwind α_U nodal values designated depending on the flux direction (see Figure

4.1) to express the normalized variable as

$$\tilde{\alpha} = \frac{\alpha - \alpha_U}{\alpha_A - \alpha_U}, \tag{4.7}$$

with this normalization, the upwind and donor values are transformed to $\tilde{\alpha}_U = 0$, and $\tilde{\alpha}_A = 1$, and the face value relation $\alpha_f = f(\alpha_D, \alpha_A, \alpha_U)$ becomes $\tilde{\alpha}_f = f(\tilde{\alpha}_D)$.

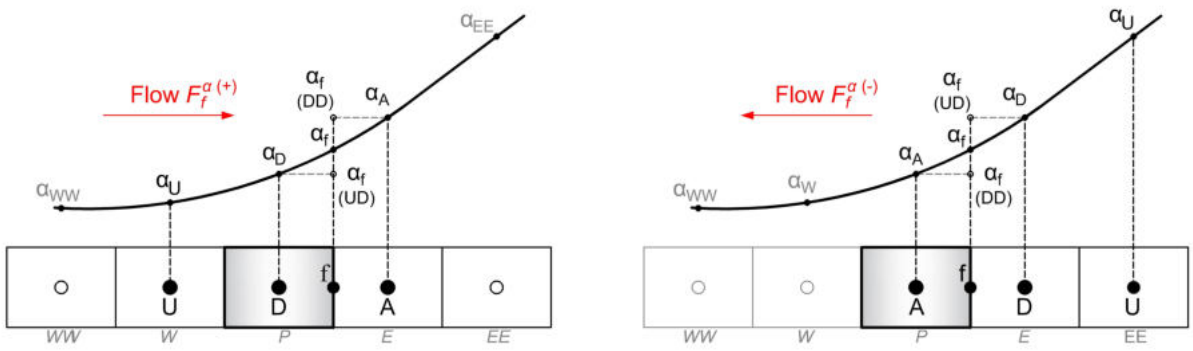


Figure 4.1.: The schematic position of D, A, and U nodes according to the flow direction in a one-dimensional domain

In the case that the nodal value U does not exist, e.g., on the block boundaries at face e when the volumetric flow F_e^{α} is negative, the upwind value is constructed with the Jasak [96] approach as

$$\alpha_U^* = \alpha_A - 2\nabla\alpha_D \cdot \mathbf{d}_{DA} \tag{4.8}$$

where $\mathbf{d}_{DA}$ is the vector between the nodes D and A. The gradient of α is computed with the CTS scheme like the pressure gradient (Eq. (3.13) instead variable p uses α) or with the Green Gauss gradient with the midpoint correction for non-orthogonal grids described in Appendix B. Then, to prevent unphysical values, the α_U is constrained by

$$\alpha_U = \min\left[\max\left(\alpha_U^*, 0\right), 1\right] \tag{4.9}$$

The NVD shown in Figure 4.2 displays the HR schemes' functional relations between the normalized face volume fraction $\tilde{\alpha}_f$ and $\tilde{\alpha}_D$. It is a useful tool to develop and analyze the nature of HR schemes rapidly. The UD line refers to the upwind differencing scheme, DD to the downwind differencing scheme, and the shaded area indicates the part of the NVD that fulfills the CBC. The schemes close to the UD line are linked with numerical diffusion but always produce a bounded solution and are stable. Whereas the schemes near the DD line so-called compressive schemes introduce a negative numerical diffusion but are unstable.

4.3. Interface capturing schemes

The interface capturing schemes are blended advection schemes between a compressive and a high-resolution diffusive scheme used to advect the volume fraction [159]. Some example of

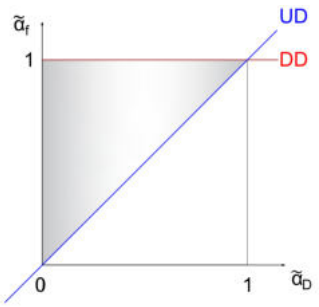


Figure 4.2.: The NVD combined with the CBC showing the region where $\tilde{\alpha}_f$ is bounded

them are: CICSAM [209], HRIC [149], IGDS [95], STACS [32], BICS [167], FBICS [204], and CUIBS [159]. These blended schemes are preferred in the multiphase community since using only compressive schemes can cause an alignment of the fluid interface with the grid [210]. While using only HR diffusive schemes deteriorates the accuracy when the flow is not orientated along a grid line due to the false diffusion [32].

In general, a blended HR scheme designed with the NVF approximates the normalized face volume fraction with a relation of the form

$$\tilde{\alpha}_f^{\text{blended HR}} = \lambda \tilde{\alpha}_f^{\text{Compressive}} + (1-\lambda)\, \tilde{\alpha}_f^{\text{Diffusive HR}}, \tag{4.10}$$

where the blending function $\lambda = f(\theta_f)$ varies between zero to one. θ_f denotes the angle between the vector normal to the interface, $\nabla \alpha_D$, and the vector $\mathbf{d}_{DA}$ and it is computed as

$$\theta_f = \arccos \left| \frac{\nabla \alpha_D \cdot \mathbf{d}_{DA}}{\|\nabla \alpha_D\| \, \|\mathbf{d}_{DA}\|} \right|. \tag{4.11}$$

To observe the interface capturing schemes' performance in complex multiphase flow problems and find a suitable scheme to deal with moving domains, the following ***six interface capturing schemes: CICSAM, HRIC, MCICSAM-W [225], MCICSAM-Z [246], FBICS, and CUIBS were implemented***. Appendix C contains their formulations in the NVF framework for uniform grids. Moreover, the interface capturing schemes originally formulated for uniform grids were modified to be used with non-orthogonal and non-uniform grids following the space normalization methodology proposed by Moukalled and Darwish in [33, 146].

4.4. Numerical implementation of the blended HR schemes

Due to the composite nature of $\tilde{\alpha}_f^{\text{blended HR}}$, it cannot be directly expressed in terms of the nodal values of the control volume P and neighbors F, which is necessary to determine the A_P^{α} and A_F^{α} coefficients and to solve Eq. (4.3) for the unknown values at the central nodes. So, several methodologies described below can be employed for the numerical implementation of the blended HR schemes.

For instance, in reference [144], the blended interface capturing scheme BICS [224] was implemented with the **deferred correction** (DC) method of Rubin and Khosla [177] to develop

a new code for simulating free surface flows around modern sailing yachts. The DC method ensembles the A-coefficients matrix using the upwind scheme, whereas the difference between the blended scheme and the upwind scheme is considered a source term. Thus, the resulting coefficient matrix is always diagonally dominant, yielding a numerically stable method. Nevertheless, the convergence rate decreases as the difference between the cell face value estimated by the upwind scheme and the blended HR scheme increases [34, 147].

On the other hand, CICSAM and its modifications, such as THOR [85], MCICSAM-W, and MCICSAM-Z, utilize the **Downwind Weighting Factor** (DWF) method developed by Leonard and Mokhtari [120]. The DWF introduces an additional factor that implicitly contains higher-order wide-stencil information, but its implementation involves only the donor and acceptor node values. As a result, this method is suitable for tridiagonal solvers. However, the diagonal coefficient A_P^{α} becomes negative when $\alpha_f > 0.5\,(\alpha_D + \alpha_A)$, which is a common scenario for all blended HR schemes when $\alpha_D > 0.5$. Consequently, this equation system leads to unphysical results for many flow configurations and requires substantial relaxation to avoid convergence problems [34]. Despite the described issue, this method is still used commonly in the multiphase community.

Another technique that overcomes the shortcomings of the DWF method and the DC method, but rarely applied in multiphase flows, is the full implicit Normalized Weighting Factor (NWF) method of Darwish and Moukalled [34]. The NWF linearizes the normalized interpolation profiles and rewrites the face value directly using the donor, acceptor, and upwind nodes. The method uses a pentadiagonal stencil, and the diagonal coefficients are always positive. As a result, the NWF is much more robust than the DWF and faster than DC methods [34]. Nonetheless, NWF is not frequently used because it requires the pentadiagonal matrix algorithm (PDMA) to solve the equations system.

A tridiagonal version of the previously described NWF method is the **Reviewed Normalized Weighting Factor** (RNWF) method presented by Chourushi [28]. This method relies on the final discretization of the normalized weighting factor method and removes the contribution of far-off nodal values from the diagonal coefficient. These terms are later added as a source term. According to the test cases studied by Chourushi, the RNWF method is faster than DC and NWF. However, for the numerical implementation of the interface capturing schemes in the context of multiphase flows, we found that the RNWF shows a similar convergence rate as the DC method and sometimes slightly altered the interface geometry. We supposed that the problem lies in introducing two explicit terms in the source term, the center point's value and the upwind point, instead of only the upwind point suggested in the original NWF method [147].

Encouraged by the numerical stability and efficiency of the NWF method, its formulation was again reviewed for use in the context of multiphase flows. As a result, a new alternative was developed within the scope of this work. This alternative is called **Modified Normalized Weighting Factor** (MNWF) method, starting from the initial formulation of the NWF method and not from its final discretized equation as the RNWF method does. The idea is only to ensemble the A-coefficient matrix with the contribution of the donor and acceptor nodes. While the contribution of the upwind node explicitly forms part of the source term. Therefore, the method results in a tridiagonal linear system. The validation of the new MNWF method, and a comparison study with the other described techniques, was published by us in the "Journal of Experimental and Computational Multiphase Flow" [135]. For that investigation, *the six interface capturing schemes: CICSAM, MCICSAM-W, MCICSAM-Z, HRIC, FBICS, and CUIBS, were numerically implemented with*

the DC, DWF, RNWF, and the new MNWF method and used to solve four test cases. Those results demonstrated that the new MNWF method requires about 5-25% fewer iterations than DC and RNWF and around 10-85% less than DWF. Thus, a similar order of accuracy of the results can be obtained with less computational time. The following section explains in detail the proposed new alternative.

4.5. The Modified Normalized Weighting Factor (MNWF) method

The formulation procedure is as follows:

First, each function of the normalized blended HR scheme is written using the NWF [34] formulation, which describes it as a linear function of the normalized donor value in the form

$$\tilde{\alpha}_f = \ell \tilde{\alpha}_D + m, \tag{4.12}$$

where ℓ represents the slope and m the intercept of each linear function in the NVD, for instance, the linearization of the interface capturing schemes CUIBS [159] expressed using the factors ℓ and m is given in Table 4.1. Appendix C contains the $[\ell, m]$ factors for some other popular blended HR schemes used to advect the volume fraction.

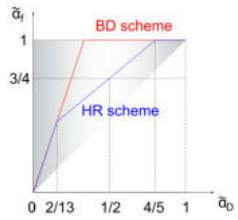
Normalized Variable Formulation (NVF)	**Linearization factors** $[\ell, m]$
	
$\tilde{\alpha}_f^{\text{BD}} = \begin{cases} 3\tilde{\alpha}_D & 0 < \tilde{\alpha}_D \leq \frac{1}{3} \\ 1 & \frac{1}{3} < \tilde{\alpha}_D \leq 1 \\ \tilde{\alpha}_D & \text{otherwise} \end{cases}$	$[\ell, m]^{\text{BD}} = \begin{cases} [3, 0] & 0 < \tilde{\alpha}_D \leq \frac{1}{3} \\ [0, 1] & \frac{1}{3} < \tilde{\alpha}_D \leq 1 \\ [1, 0] & \text{otherwise} \end{cases}$
$\tilde{\alpha}_f^{\text{HR}} = \begin{cases} 3\tilde{\alpha}_D & 0 < \tilde{\alpha}_D \leq \frac{2}{13} \\ \frac{5}{6}\tilde{\alpha}_D + \frac{1}{3} & \frac{2}{13} < \tilde{\alpha}_D \leq \frac{4}{5} \\ 1 & \frac{4}{5} < \tilde{\alpha}_D \leq 1 \\ \tilde{\alpha}_D & \text{otherwise} \end{cases}$	$[\ell, m]^{\text{HR}} = \begin{cases} [3, 0] & 0 < \tilde{\alpha}_D \leq \frac{2}{13} \\ \left[\frac{5}{6}, \frac{1}{3}\right] & \frac{2}{13} < \tilde{\alpha}_D \leq \frac{4}{5} \\ [0, 1] & \frac{4}{5} < \tilde{\alpha}_D \leq 1 \\ [1, 0] & \text{otherwise} \end{cases}$
$\tilde{\alpha}_f^{\text{CUIBS}} = \lambda \tilde{\alpha}_f^{\text{BD}} + (1 - \lambda)\, \tilde{\alpha}_f^{\text{HR}}$	$[\ell, m]^{\text{CUIBS}} = \lambda_f [\ell, m]^{\text{BD}} + (1 - \lambda_f)\, [\ell, m]^{\text{HR}}$

Table 4.1.: The Normalized Variable formulation of the CUIBS scheme and its corresponding $[\ell, m]$ factors. BD denotes the compressive Bounded Downwind scheme, HR the diffusive high-resolution scheme and $\lambda_f = |\cos\theta_f|^4$ is the blended function

Second, to ensure numerical consistency, the m factor of the blended HR scheme is corrected with the CBC condition. $\tilde{\alpha}_f^{\text{blended-HR}}$ is explicitly calculated using the blended HR scheme and then bounded as

$$\min\left(\tilde{\alpha}_D, 1\right) \leq \tilde{\alpha}_f^{\text{blended-HR}} \leq \max\left(\tilde{\alpha}_D, 1\right). \tag{4.13}$$

This bounded value is used to corrected m according to

$$m^{\text{blended-HR}} = \tilde{\alpha}_f^{\text{blended-HR}} - \ell^{\text{blended-HR}} \tilde{\alpha}_D. \tag{4.14}$$

Note that when $[\ell, m]^{\text{blended-HR}} = [0, 1]$, the coefficient A_P^{α} becomes zero, and the A-coefficient matrix will no more be dominant, which leads to numerical instabilities [34]. In this case, the strategy introduced in the RNWF [28] methodology is used instead of the original NWF strategy

as it proved to be more numerically stable. Thus,

$$[\ell, m]^{\text{blended-HR}} = [L, 1 - L\tilde{\alpha}_D]\,. \tag{4.15}$$

where L is the factor ℓ of an interval of the diffusive HR scheme to preserve stability.

Third, the final $\tilde{\alpha}_f^{\text{blended-HR}}$ is also assumed to take the form of the linear relation 4.12, which is denormalized and rewritten as

$$\frac{\alpha_f - \alpha_U}{\alpha_A - \alpha_U} = \ell \frac{\alpha_D - \alpha_U}{\alpha_A - \alpha_U} + m, \tag{4.16}$$

yielding

$$\alpha_f = \ell\left(\alpha_D - \alpha_U\right) + m\left(\alpha_A - \alpha_U\right) + \alpha_U = \ell\alpha_D + m\alpha_A + \left(1 - \ell - m\right)\alpha_U. \tag{4.17}$$

Finally, defining the volume fraction on the face with Eq. (4.17), the convective term is written in terms of the nodal values as

$$\begin{aligned} F_f^\alpha \alpha_f = {} & \max(0, F_f^\alpha)\left[\ell_f \alpha_P + m_f \alpha_F + \left(1 - \ell_f - m_f\right)\alpha_U^+\right] \\ & - \max(-F_f^\alpha, 0)\left[\ell_f \alpha_F + m_f \alpha_P + \left(1 - \ell_f - m_f\right)\alpha_U^-\right]. \end{aligned} \tag{4.18}$$

In the NWF approach, Eq. (4.18) is solved fully implicitly. The acceptor, donor, and far nodes (α_U^+ and α_U^-) are part of the A-coefficients, which results in a pentadiagonal system of equations. On the contrary, in the MNWF method, only the terms related to the donor and acceptor nodal values are used to form the A-coefficients matrix, while the last term involving the far node U is explicitly determined and added directly to the source term, as is shown in reference [147]. It leads to a tridiagonal A-coefficients matrix.

4.5.1. The algebraic VoF equation resulting from the MNWF method

Using Eq. (4.18) in the context of the MNWF method, the coefficient of the algebraic equation (4.3), written here again for convenience,

$$\left(A_{P(time)}^\alpha + A_P^\alpha\right)\alpha_P = \sum_{F(P)} A_F^\alpha \alpha_F + S_P^\alpha, \tag{4.19}$$

take the form,

$$\begin{aligned} A_F^\alpha &= -\left[\max\left(0, F_f^\alpha\right) m_f - \max\left(-F_f^\alpha, 0\right)\ell_f\right], \\ A_P^\alpha &= \sum_{f(P)}\left[\max\left(0, F_f^\alpha\right)\ell_f - \max\left(-F_f^\alpha, 0\right) m_f\right], \\ S_f^\alpha &= \left(1 - \ell_f - m_f\right)\left[\max\left(0, F_f^\alpha\right)\alpha_U^{+(m)} - \max\left(-F_f^\alpha, 0\right)\alpha_U^{-(m)}\right], \\ S_P^\alpha &= S_{P(time)}^\alpha - \sum_{f(P)} S_f^\alpha, \end{aligned} \tag{4.20}$$

where the superscript m denotes the last available value.

For example, applying the MNWF method to define the algebraic equation for the one-dimensional domain of Figure 4.1 results in

$$\left(A^{\alpha}_{P(time)} + A^{\alpha}_{P}\right)\alpha_P = \sum_{F(P)} A^{\alpha}_{F}\alpha_F + S^{\alpha}_{P}, \quad (4.21) \tag{4.21}$$

where

$$\begin{aligned}
A^{\alpha}_{E} &= -\left[\max\left(0, F^{\alpha}_{e}\right) m_e - \max\left(-F^{\alpha}_{e}, 0\right) \ell_e\right] \\
A^{\alpha}_{W} &= -\left[\max\left(0, F^{\alpha}_{w}\right) m_w - \max\left(-F^{\alpha}_{w}, 0\right) \ell_w\right] \\
A^{\alpha}_{P} &= A^{\alpha}_{P(time)} + \left[\max\left(0, F^{\alpha}_{e}\right) \ell_e - \max\left(-F^{\alpha}_{e}, 0\right) m_e\right] + \left[\max\left(0, F^{\alpha}_{w}\right) \ell_w - \max\left(-F^{\alpha}_{w}, 0\right) m_w\right] \\
S^{\alpha}_{e} &= \left(1 - \ell_e - m_e\right)\left[\max\left(0, F^{\alpha}_{e}\right) \alpha^{m}_{W} - \max\left(-F^{\alpha}_{e}, 0\right) \alpha^{m}_{EE}\right] \\
S^{\alpha}_{w} &= \left(1 - \ell_w - m_w\right)\left[\max\left(0, F^{\alpha}_{w}\right) \alpha^{m}_{WW} - \max\left(-F^{\alpha}_{w}, 0\right) \alpha^{m}_{E}\right] \\
S^{\alpha}_{P} &= S^{\alpha}_{P(time)} - \left(S^{\alpha}_{e} + S^{\alpha}_{w}\right)
\end{aligned} \tag{4.22}$$

Similarly to the algebraic momentum equation (3.34), the algebraic VoF equation, Eq. (4.19), is iteratively solved and can be implicitly under-relaxed with a parameter $0 < \gamma^{\alpha} < 1$, which leads to

$$\underbrace{\frac{\left(A^{\alpha}_{P(time)} + A^{\alpha}_{P}\right)}{\gamma^{\alpha}}}_{\tilde{A}^{\alpha}_{P}} \alpha^{n+1}_{P} = \sum_{f(P)} A^{\alpha}_{F}\alpha^{n+1}_{F} + \underbrace{S^{\alpha,n}_{P} + \frac{1-\gamma^{\alpha}}{\gamma^{\alpha}} A^{\alpha}_{P}\alpha^{m}_{P}}_{\tilde{S}^{\alpha,n}_{P}}. \tag{4.23}$$

Eq. (4.23) is ensembled for all the control volumes that are not at the boundaries of the problem domain. The coefficients are modified for the boundary CVs, as is shown in Section 4.6. The resulting linear system of N equations is solved with the incomplete LU decomposition method. Then, since sometimes non-physical volume fraction values can occur [209], these values are reset to zero or one using the following constrain

$$\alpha_P = \min\left[\max\left(\alpha_P, 0\right), 1\right]. \tag{4.24}$$

Additionally, following the recommendation of Hirt [82] to avoid spurious results, the α field is also corrected by

$$\alpha_P = \begin{cases} 0 & \alpha_P < \varepsilon_{\alpha} \\ 1 & \alpha_P < (1 - \varepsilon_{\alpha}) \end{cases}, \quad \text{with} \quad \varepsilon_{\alpha} = 10^{-6}. \tag{4.25}$$

Of course, these changes of the nodal values introduce a conservation error [210]. However, according to Hirt [82], the volume errors after hundreds of cycles are typically a fraction of one percent of the total fluid volume. In the cases here studied, similar negligible errors in conservation have been observed, and instead, the application of these constraints speed up the numerical resolution.

Finally, the central nodal values of the boundary CVs are copied to the boundaries, and the next iteration is carried out until the convergence criterion given by Eq. (3.60) is reached.

4.6. Boundary conditions for the VoF equation

The treatment of the *Inlet, Outlet, Wall,* and *Symmetry* boundary conditions for the pure convection VoF equation are now detailed. The implementation follows the methodology published in [147] for pure convective problems.

Let P be the center of a boundary CV with one boundary face of centroid b and surface vector $\mathbf{n}S_b$ pointing outward, as shown in Figure 4.3. As before, the discretization process over CV P in a multidimensional domain yields

$$A^{\alpha}_{P(time)}\alpha_P = -\sum_{f(P)} F^{\alpha}_f \alpha_f + S^{\alpha}_{P(time)}, \tag{4.26}$$

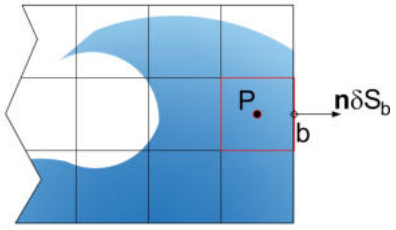


Figure 4.3.: Boundary control volume P with one boundary face b

The convective fluxes on the interior faces are discretized as described in the previous Section 4.5.1, and independent of the boundary condition type, the convective boundary flux can be directly written using the boundary face centroid values as

$$F^{\alpha}_b \alpha_b = \left(\mathbf{v}^n_b \cdot \mathbf{n}\delta S^{n+1}_b - \mathbf{v}^{n+1}_{g,b} \cdot \mathbf{n}\delta S^{n+1}_b\right)\alpha_b. \tag{4.27}$$

Hence, the discretized equation of the boundary CV takes the form

$$A^{\alpha}_{P(time)}\alpha_P = -\sum_{f(P)} F^{\alpha}_f \alpha_f - F^{\alpha}_b \alpha_b + b^{\alpha}_{P(time)}, \tag{4.28}$$

where subscript f now relates only to interior faces and subscript b to the boundary face. The specification of boundary conditions requires either specifying the unknown boundary value α_b or the boundary flux F^{α}_b.

At an **inlet** boundary, the velocity value is prescribed. Thus, the inlet convective flux is also known, and the boundary flux is addressed as a source term. With this change, the algebraic equation for the boundary CV is given by

$$\left(A^{\alpha}_{P(time)} + A^{\alpha}_P\right)\alpha_P = \sum_{F(P)} A^{\alpha}_F \alpha_F + S^{\alpha}_P, \tag{4.29}$$

where

$$
\begin{aligned}
A_F^\alpha &= -\left[\max\left(0, F_f^\alpha\right) m_f - \max\left(-F_f^\alpha, 0\right) \ell_f\right], \\
A_P^\alpha &= \sum_{f(P)} \left[\max\left(0, F_f^\alpha\right) \ell_f - \max\left(-F_f^\alpha, 0\right) m_f\right], \\
S_f^\alpha &= (1 - \ell_f - m_f) \left[\max\left(0, F_f^\alpha\right) \alpha_U^{+(n)} - \max\left(-F_f^\alpha, 0\right) \alpha_U^{-(n)}\right], \\
b_P^\alpha &= b_{P(time)}^\alpha - \sum_{f(P)} S_f^\alpha - F_b^\alpha \alpha_b.
\end{aligned}
\tag{4.30}
$$

Here, F only denotes the interior neighboring control volumes of P, and f refers to the boundary CV's interior faces.

At the **outlet** from the domain, a fully developed flow is assumed, which is equivalent to assuming that the normal gradient to the face is zero, $(\partial\alpha/\partial\mathbf{n}\,)_b = 0$. This condition is fulfilled by assigning $\alpha_b = \alpha_P$. As a result, the coefficient for the algebraic equation, Eq. (4.29), for an outlet CV P becomes

$$
\begin{aligned}
A_F^\alpha &= -\left[\max\left(0, F_f^\alpha\right) m_f - \max\left(-F_f^\alpha, 0\right) \ell_f\right], \\
A_P^\alpha &= \sum_{f(P)} \left[\max\left(0, F_f^\alpha\right) \ell_f - \max\left(-F_f^\alpha, 0\right) m_f\right] + F_b^\alpha, \\
S_f^\alpha &= (1 - \ell_f - m_f) \left[\max\left(0, F_f^\alpha\right) \alpha_U^{+(n)} - \max\left(-F_f^\alpha, 0\right) \alpha_U^{-(n)}\right], \\
b_P^\alpha &= b_{P(time)}^\alpha - \sum_{f(P)} S_f^\alpha.
\end{aligned}
\tag{4.31}
$$

Again, the subscript F and f refer to the interior neighboring control volumes of P and its interior faces, respectively.

For a **wall** boundary condition, since the normal velocity at a wall is zero, the convection flux is zero. So, the contribution of the boundary face to the A-coefficients and source term is also zero. Only the interior faces build the algebraic equation (4.29) with the coefficient defined with Eq. (4.20).

Finally, at a **symmetry** plane, no flow crosses the boundary. In other words, the convection flux normal to a symmetry plane is also zero. Consequently, it is treated similarly to the wall boundary condition.

4.7. Control of the Courant number condition

The accuracy of the solution resulting from the interface capturing schemes depends on keeping the local Courant numbers in a range for which the scheme can capture the interface without introducing excessive numerical diffusion [83, 118].

Here, the Courant number is controlled during the simulation by an automatic time-step refinement. One must prescribe Co_{mean} and Co_{max} that refer to the mean and the maximum local Courant number allowed for the calculation. The minimum Co number denoted by Co_{min} is by default 0.01.

Before starting the time-step computation, the solver calculates the local Courant numbers given by Eq. (4.6) for all the domain, finds the maximum value, and defines a factor H given by

$$\frac{Co_P^{\max}}{\Delta t^{old}} = \frac{\sum_{f(P),out} \left| F_f^{\alpha} \right|}{\delta V_P} = H, \tag{4.32}$$

where Δt^{old} represents the current time-step.

In the case of

$$Co_P^{\max} < Co\_\min \quad \text{or} \quad Co_P^{\max} > Co\_\max, \tag{4.33}$$

the time-step is automatically modified to satisfy the mean local Co allowed for the calculation by

$$\Delta t^{new} = \frac{Co\_mean}{H}. \tag{4.34}$$

It is worth pointing out that the time-step refinement starts after completing the first ten time-step computations to avoid extreme variations that could cause an initially non-convergent velocity field.

4.8. Two-fluid flows solution algorithm

The solution algorithm employed for any multifluid case is a sequential one. The computation starts from the prescribed initial condition, which defines the component fluids' current distribution and physical properties.

Figure 4.4 depicts the solution algorithm for a two-fluid problem in a fixed domain, which can be summarized with the following steps:

1. Assume that the last available solution of the variables at t_n is the current solution at t_{n+1}.
2. Compute the local Co number with Eq. (4.6) for the whole domain and find the maximum Co.
3. If the adaptive time-step is active, compute the new time-step to satisfy the Co condition with Eq. (4.34).
4. Increment time with $t_{n+1} = t_n + \Delta t$
5. Advect α field with one of the interface capturing schemes of Appendix C implemented with the MNWF method and ensemble the A-coefficients and Source term of Eq. (4.22)
6. Solve the volume fraction equation (4.23) to find α
7. Correct the non-physical α-values

   α converges? **no**: go to step **8**, **yes**: go to step **9**
8. Compute the $\nabla \alpha_P$ with the CST method, or apply the Green Gauss gradient with the midpoint correction for non-orthogonal grids given in Appendix B. Then, go to step **5**

9. Update the nodal physical properties, density, and viscosity, with Eq. (2.8).
10. Compute the weighting factors n_f using Eq. (3.39) and update the density and viscosity on the face-centers applying Eq. (3.37).
11. Calculate the surface tension forces
12. Compute the predictor-corrector enhanced SIMPLE algorithm by PRIME steps described in Section 3.8 to solve the velocity, pressure, and mass flow rate fields.
13. Update the volumetric flow rates given by Eq. (4.2)
14. Go to step **1**

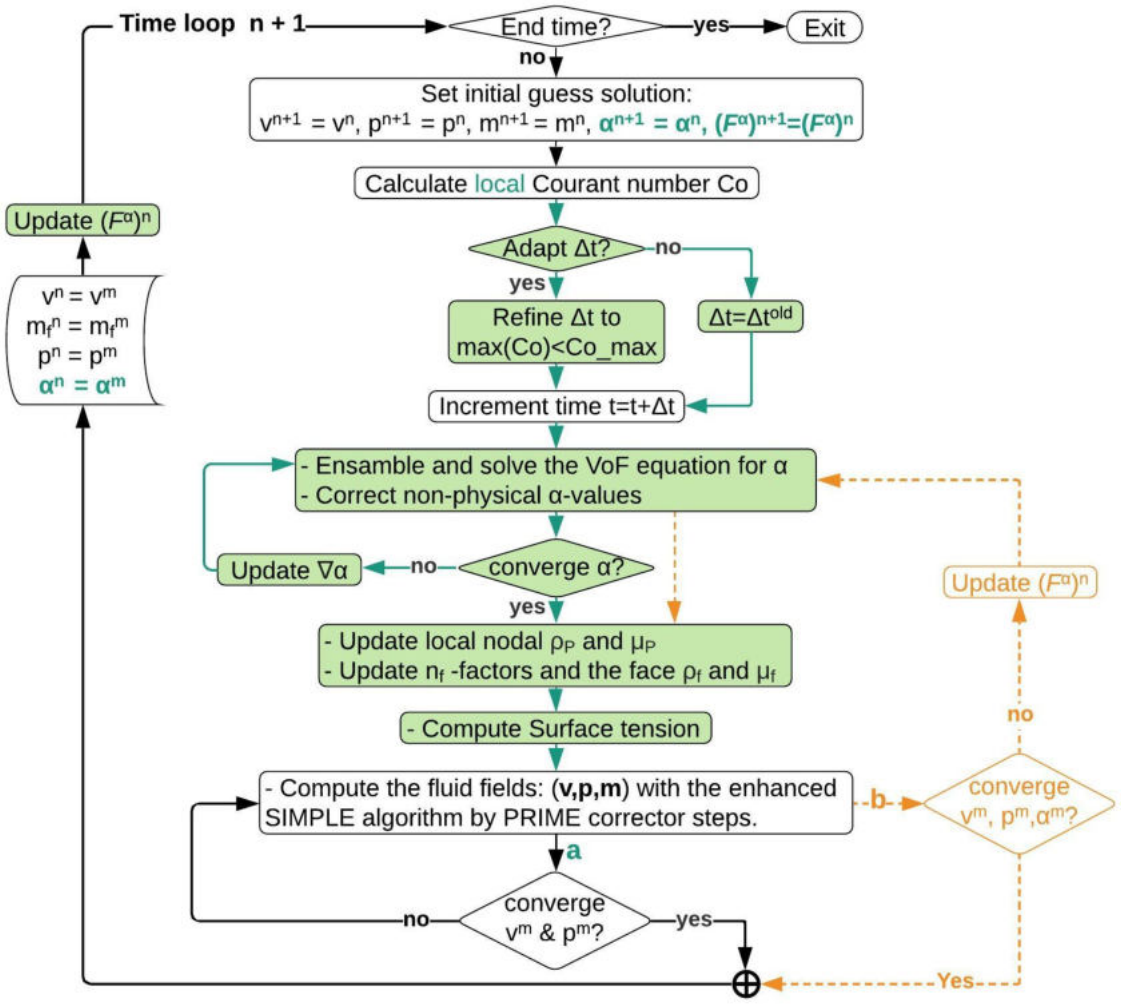


Figure 4.4.: Flow chart of solution algorithm for two-fluids flow problems in a fixed domain. The green color highlights the new implementations

Generally, I recommend the above sequence. However, note that there is an orange dashed line in the flowchart, indicated as b. It includes the solution of the VoF equation into the pressure-velocity coupling algorithm. Thus, the volume fraction field is updated as the same frequency as the velocity and pressure field. In some scenarios, the changes in velocity are very significant between each iteration. It can considerably influence the advection of the free surface. Therefore, the volume fraction needs to be updated as often as the velocity to obtain accurate results. In these cases, using loop-b, the individual iteration loop of α, given by step 8, is skipped. It is worth mentioning that loop-b should be used only for the problems that the standard sequence (loop-a in the flowchart) gives not a good result even when a smaller time-step is used. The frequent updating of the volume fraction field introduces excessive changes in physical properties, according to the analysis shown by us in [134]. These continuous changes decrease the convergence rate of the pressure-velocity coupling algorithm. The computational time can rise by almost 60% in comparison to using loop-a.

The last remarks concern temporal discretization. The first-order implicit Euler scheme, while computationally stable and efficient, introduces substantial numerical diffusion. In contrast, the BDF2 time scheme is notably more accurate than implicit Euler [134] but can lead to over/under-shoots with large time-steps, as it is not bounded [32]. Usually, the BDF2 scheme is used in this work.

4.9. Performance analysis of six interface capturing schemes at different Courant numbers

This section aims to evaluate the performance of the six interface capturing schemes listed in Appendix C (CICSAM, MCICSAM-W, MCICSAM-Z, HRIC, FBICS, and CUIBS) at high Courant number conditions. The schemes are numerically implemented following the MNWF methodology described in Section 4.5 since this new method has superior performance to other implementation methods (see [135]).

For this part, two test cases are considered: the slotted circle's advection in a rotational flow field introduced in [244] and the advection of a circle in a shear flow presented in [179]. Both have simple exact solutions and are frequently used in the multiphase community to validate the advection schemes' performance dealing with a non-uniform distribution of the Courant number and a considerable interface deformation, respectively [159]. The initialization of the volume fraction field for these cases is illustrated in Figure 4.5.

The performance is measured by the efficiency and the accuracy of solving a problem by the advection scheme. The efficiency is related to the computational effort; in other words, the total number of iterations required to achieve the convergence criterion (for these cases $\varepsilon = 5 \times 10^{-3}$) during the simulation without under-relaxation. Whereas the accuracy of the results is verified using the root mean square (RMS) defined as

$$error = \sqrt{\frac{\sum_{i}^{N} \left(\alpha_i^{numeric} - \alpha_i^{analytic}\right)^2}{N}}, \tag{4.35}$$

where N is the total number of CVs.

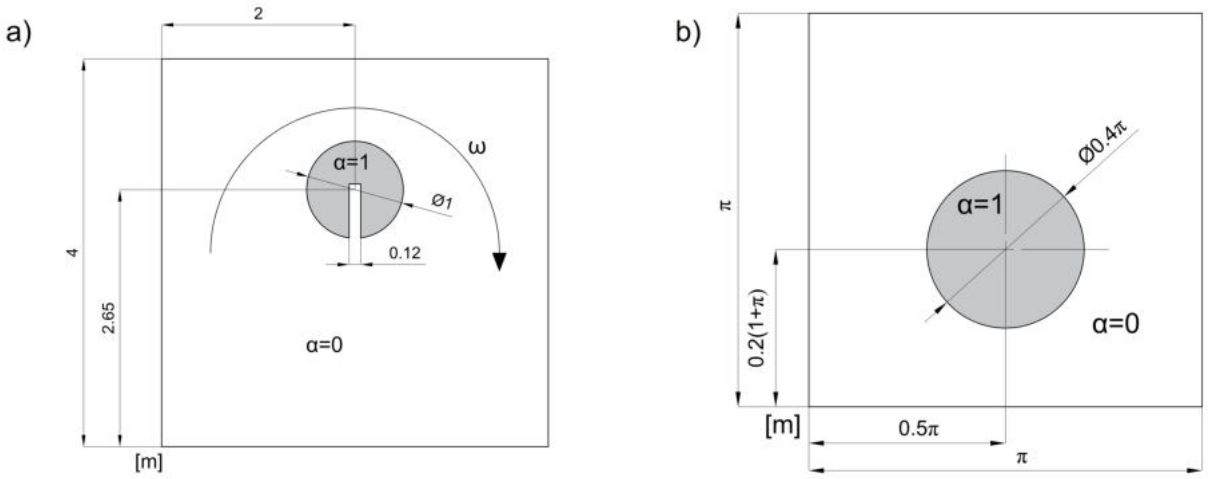


Figure 4.5.: Initial configuration for a) the slotted circle test case and b) the circle in a shear flow

4.9.1. Advection of a Slotted Circle in a rotational flow

A circle with a diameter of 1 m and a slot of width 0.12 m and depth 0.62 m is centered at (2, 2.65) m of a square 4×4 m^2 domain and exposed to a clockwise circular velocity given by

$$u = -\omega\,(y - y_0)\,, \quad v = \omega\,(x - x_0) \tag{4.36}$$

where $(x_0, y_0) = (2, 2)$ is the center of the rotation and $\omega = 0.5$ rad/s is the constant angular velocity. The slotted circle completes a revolution in $2\pi/\omega$ s. The problem is discretized with a structured grid of 200×200 square control volumes, and it is solved for five different time-step sizes, which produce a maximum Courant number (Co) of 0.4, 0.6, 0.8, and 1.0 at point (2, 2.15) m [32]. The maximum number of iterations per time-step is set to 20.

The clustered column charts in Figure 4.7 present a) the total number of iterations and b) the accuracy of the solution of each interface capturing scheme at different maximum Courant numbers when the slotted circle completes two rotations. The difference in the efficiency of the schemes increases with the Co. Thus, at the lowest $Co = 0.4$, the total number of iterations performed by each scheme is almost the same, while at higher Co numbers, they differ by around 10% to 30%, with HRIC being the fastest to reach the convergence, whereas FBICS is the slowest. In contrast, the accuracy at each Co varies according to the interface capturing schemes used and its level of independence from the Co number. We can observe that CICSAM and CUIBS significantly deteriorate the flow interface definition when the Co increases, whereas the other schemes' accuracy, is only slightly affected. The precision difference can be better visualized in Figure 4.6, which shows the contour plots at Courant number 0.4, 0.6, 0.8, and 1.0 of the six studied schemes.

In this test case, the most accurate scheme for $Co < 0.8$ is CUIBS, and for $Co > 0.8$, FBICS and MCICSAM-Z are good candidates. On the other hand, HRIC is the fastest but suffers from diffusion independent from the Co number.

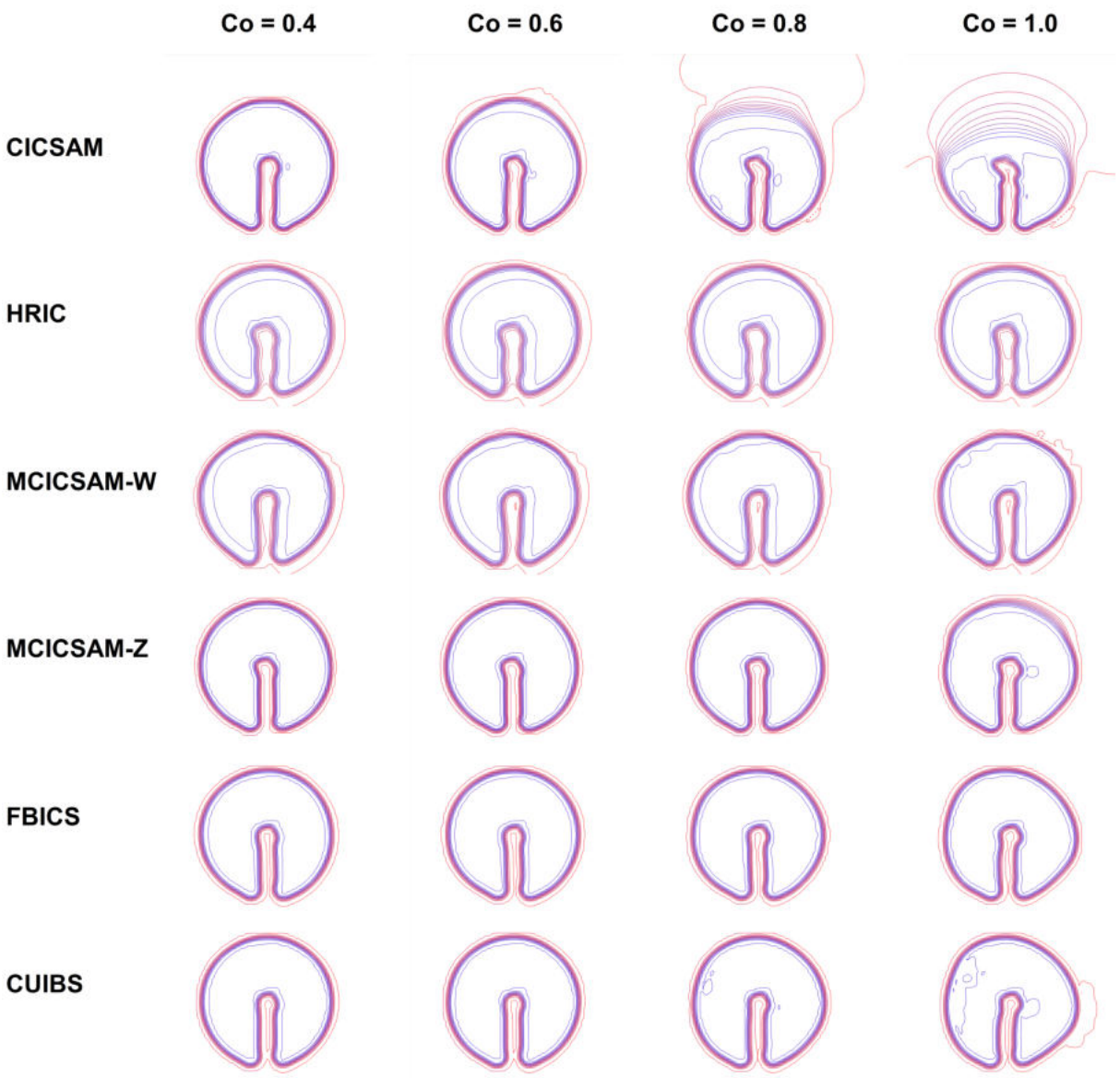


Figure 4.6.: Contour plots of the slotted circle problem after two complete rotation at different Courant number solving with six interface capturing schemes

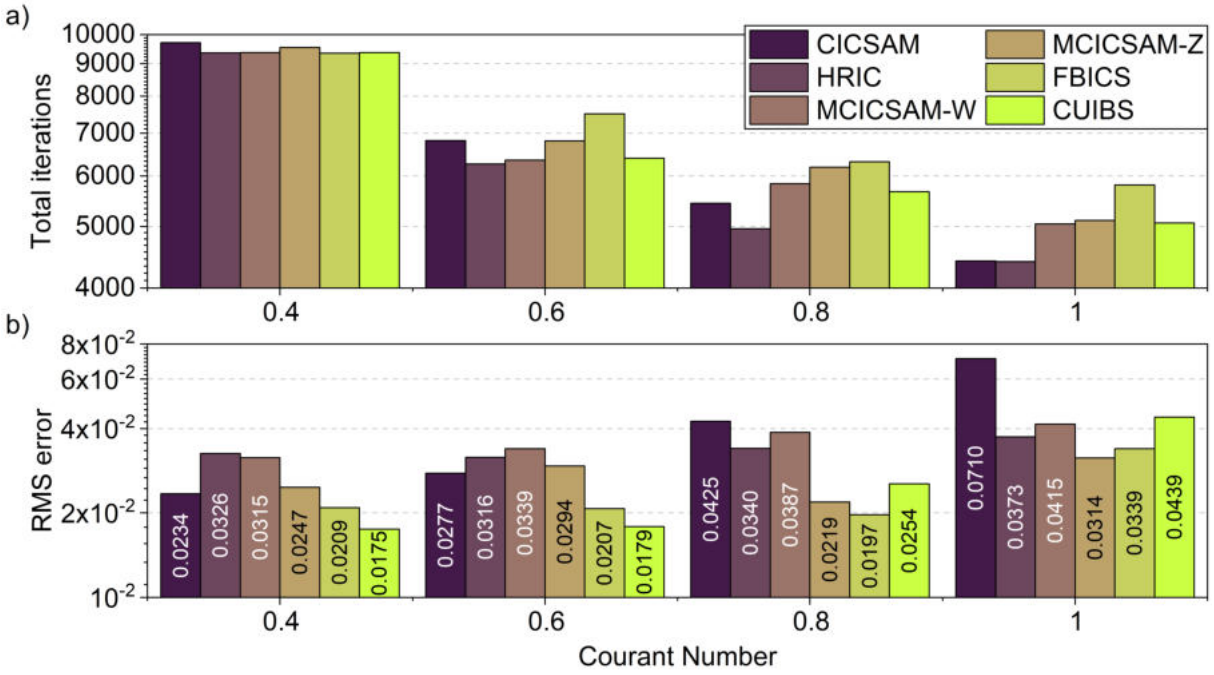


Figure 4.7.: a) The total number of iterations and b) the RMS error after two rotations of the slotted circle for six interface capturing schemes

4.9.2. Advection of a circle in a shear flow

The volume fraction field is initialized, as is shown in Figure 4.5 b. A circle of 0.2π m diameter with its center at $(0.5\pi, 0.2(1+\pi))$ m filled with phase one is in a square domain of phase two. The two-phase configuration is exposed to a shear flow field described by,

$$u = \sin(x)\cos(y), \quad v = -\cos(x)\sin(y) \tag{4.37}$$

where $(x, y) \in [0, \pi]$. The domain is discretized with a uniform structured mesh consisting of 160×160 cells. For observing the interface capturing schemes' performance in the presence of interface deformation, the simulation is firstly run for 2000 time-steps using the velocity defined in Eq. (4.37). Then, the flow is reversed, and the simulation is rerun for 2000 time-steps. Hence, the interface should return to its initial shape. For this study, two local Courant numbers are tested, $Co = 0.5$ and $Co = 0.8$.

Figure 4.9 summarizes a) the total number of iterations required to achieve convergence during the whole simulation and b) the RMS error of the results produced by each interface capturing scheme. The solver is set to perform a maximum of 10 iterations per time-step for all the runs. In the case of $Co = 0.5$, MCICSAM and FBICS are the most time-consuming. They performed almost 30% more iterations than HRIC, the fastest scheme. However, their approximations are more exact, and the RMS error between the final interface shape and the initial shape are 0.0433 and 0.0484, respectively. In the case of $Co = 0.8$, the high degree of interface deformation notably reduces the efficiency of MCICSAM-W and CUIBS. They need around 100% more iterations than

CICSAM or HRIC. At this high Co condition, the schemes that better return the interface to its initial shape are FBICS and CUIBS. The first is also an excellent combination of numerical efficiency and accuracy. The contour plots after 2000 forward and 2000 backward steps for the two considered Courant numbers are presented in Figure 4.8.

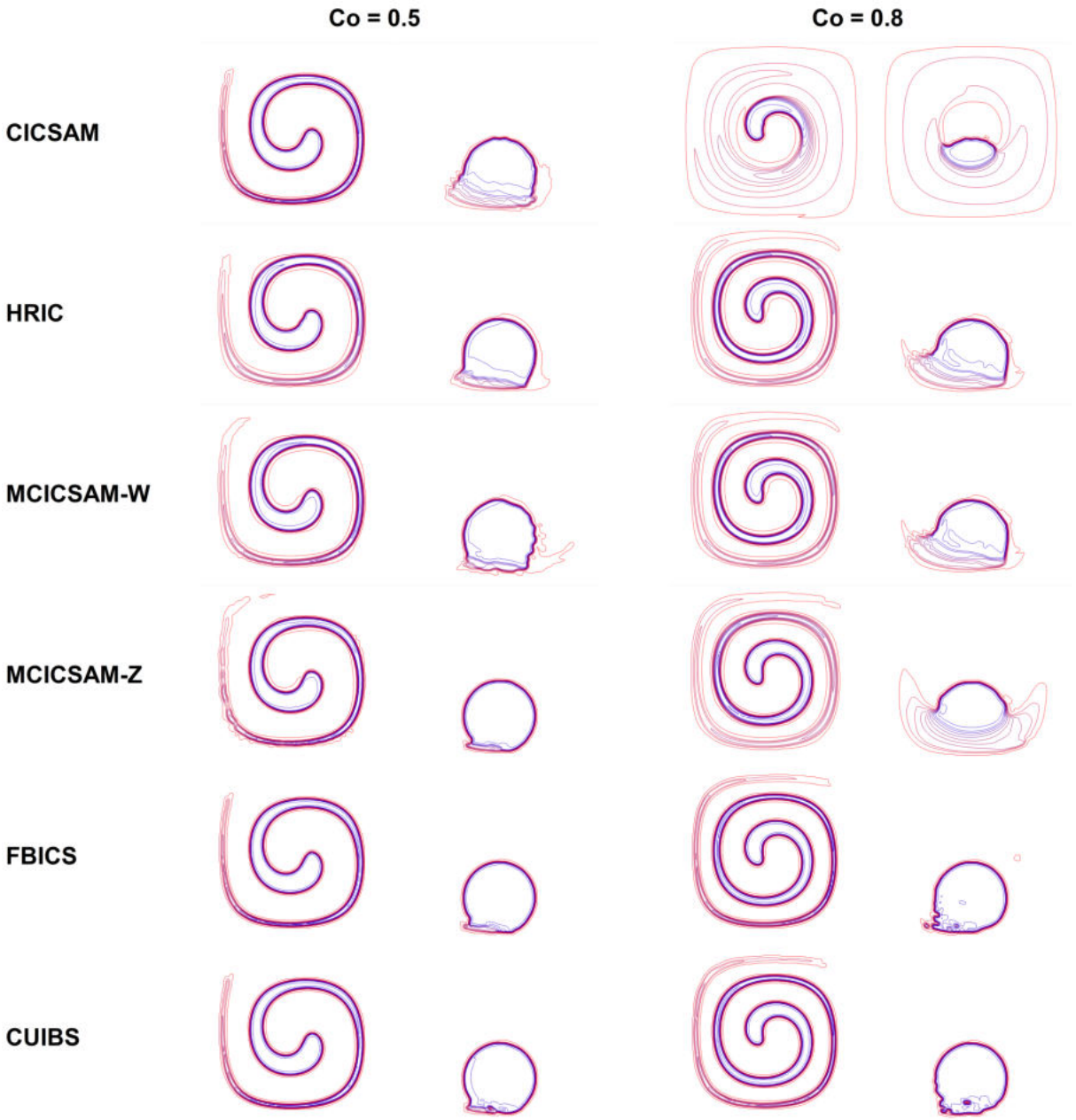


Figure 4.8.: Contour plots of the circle in a shear flow after 2000 forward steps, followed by 2000 backward steps for six interface capturing schemes at two Co numbers

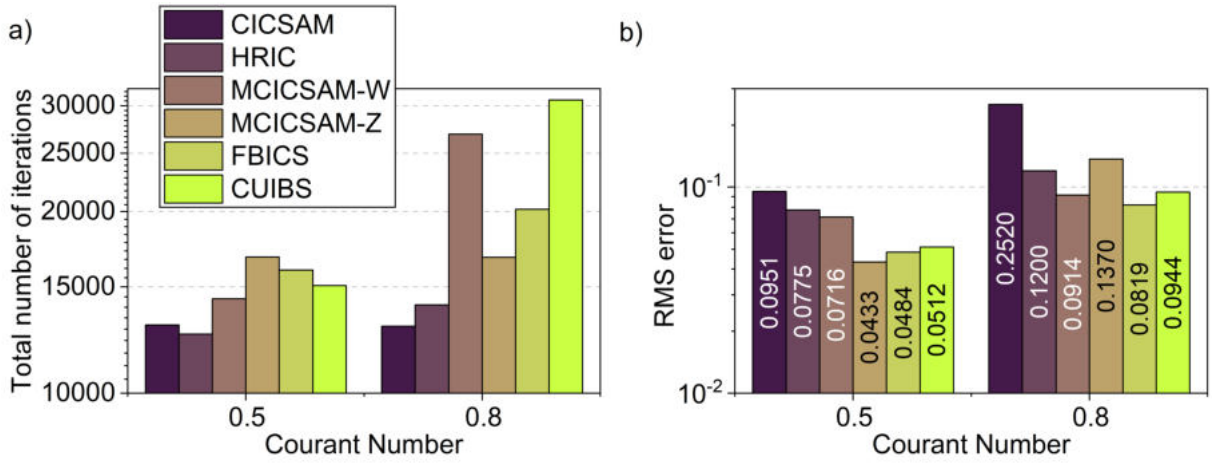


Figure 4.9.: a) The total number of iterations and b) the RMS error after 2000 forward and 2000 backward steps of the circle in shear flow test case at two high Co numbers

Conclusions:

These two cases that address the volume fraction's advection in a non-uniform distribution of the Courant number and a considerable interface deformation indicate that CICSAM and its variants and HRIC tend to be more diffusive schemes than FBICS and CUIBS. The more suitable schemes for working at medium-high Courant number conditions are MCICSAM-Z, FBICS, and CUIBS. However, unfortunately, the six schemes considerably smear out the interface at high Courant numbers ($Co > 0.8$), which causes a deterioration of the sharpness or nonphysical changes of the flow interface.

4.10. Validation of the two-fluid flow solver

The following test cases aim to validate the implementation in the FASTEST code of the two-fluid flow solution algorithm and observe its applicability to compute the most common free surface phenomena such as sloshing, bubble dynamics, and breaking waves. For this part, the interface capturing schemes used are MCICSAM-Z, FBICS, and CUIBS due to their better behavior to define sharp interfaces at medium-high Co numbers.

4.10.1. 3D Sloshing in a rectangular tank

The test case consists of a 3D sloshing of an inviscid fluid in a tank with the dimension of L×L×H and a mean flow depth of D where $L = 20$ m, $H = 1.5$ m and $D = 1$ m (see Figure 4.10). The initial free surface elevation is a Gaussian shape defined as

$$y_0(x, z) = D + \eta_0(x, z), \tag{4.38}$$

where η_0 is the initial free surface displacement above the calm water level D given by

$$\eta_0(x,z) = H_0 \exp\left\{\kappa\left[\left(x-\frac{L}{2}\right)^2+\left(z-\frac{L}{2}\right)^2\right]\right\}. \tag{4.39}$$

Let H_0 be the initial hump height equal to 0.1 m and κ the magnification factor of value 0.25 as are set in [99].

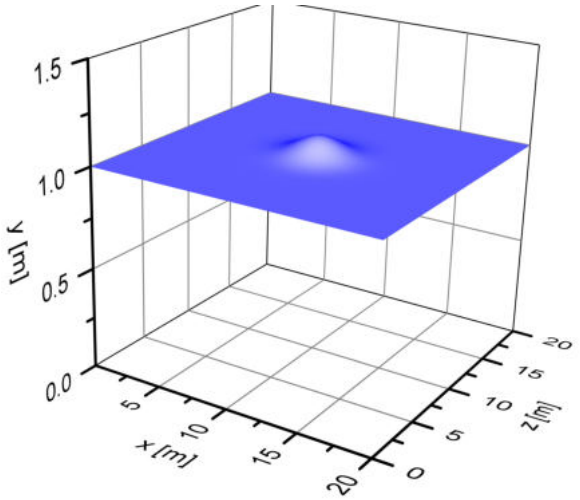


Figure 4.10.: Schematic description of the domain and the initial condition of the free surface. The initial hump height is 0.1 m

For the above initial free surface shape, Wei and Kirby [229] developed the following linear analytic solution of the free surface displacement

$$\eta(x,z,t) = \sum_{n=0}^{\infty}\sum_{m=0}^{\infty}\left[C_{nm}\exp\left(-\sqrt{-1}\omega_{mm}t\right)\cos\left(\frac{n\pi x}{L}\right)\cos\left(\frac{n\pi z}{L}\right)\right], \tag{4.40}$$

with

$$C_{nm} = \frac{4}{(1+\delta_{n0})(1+\delta_{m0})L^2}\int_0^L\int_0^L \eta_0(x,z)\cos\left(\frac{n\pi x}{L}\right)\cos\left(\frac{n\pi z}{L}\right)dxdz. \tag{4.41}$$

Each (n,m) mode has a natural frequency defined by an equation of the form

$$\omega_{nm} = g\left(\frac{\pi}{L}\right)^2\left(n^2+m^2\right)\tanh\left[\left(\frac{\pi}{L}\right)^2\left(n^2+m^2\right)D\right]. \tag{4.42}$$

The acceleration due to gravity is $g = 9.81$ m/s^2 and is directed in the y-direction.

The computational domain consists of $200 \times 48 \times 200$ CVs distributed biexponentially in the x, y, and z-direction. In the x and z-direction, the grid space decreases from 0.3 in the walls' vicinity to 0.011 m in the tank's center. Whereas in the y-direction, the CVs are clustered near $y = 1$ m.

The minimum is $\Delta y = 0.011$ m and the maximum $\Delta y = 0.095$ m. All the boundaries are defined as free-slip boundaries. The initial velocity and pressure fields are zero in the whole domain. The liquid and gas densities are set to 1000 kg/m^3 and 1 kg/m^3, respectively, and the viscosities are neglected.

The discretization in time uses the BDF2 scheme with a constant time-step $\Delta t = 0.001$ s. The momentum equation's convective term is handled with the QUICK scheme, whereas for the VoF equation, the MCICSAM-Z, FBICS, and CUIBS advection schemes are tested. For this case, the loop-b of the solution algorithm (including the VoF equation solution in the pressure-velocity coupling loop) is applied. This alternative results in better agreement with the analytical solution and not significantly different from loop-a in performance. The pressure-velocity coupling algorithm is set to perform a maximum of 50 iterations with one PRIME step (PISO mode) per each time-step. No under-relaxation is used, and the convergence tolerance for all the variables is 10^{-5}.

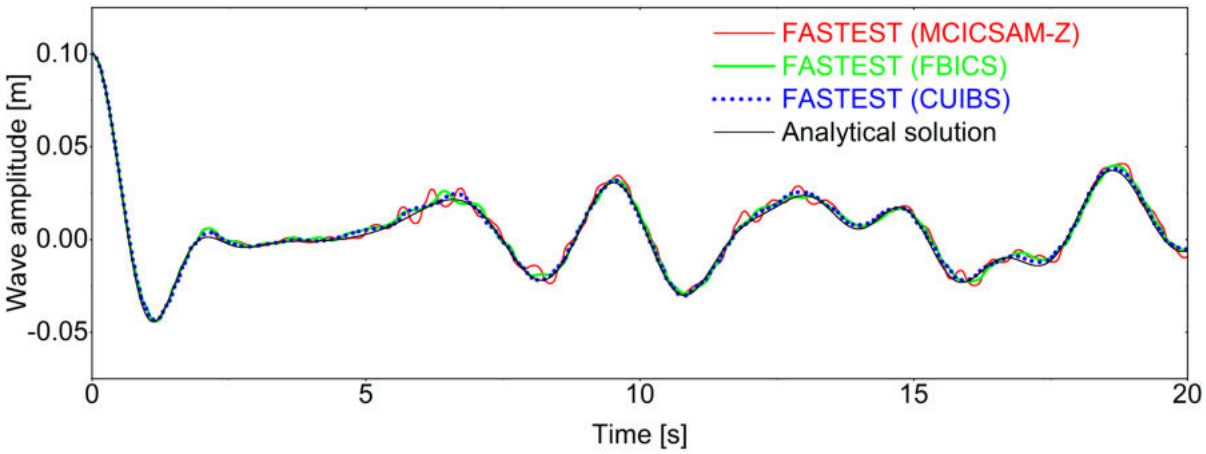


Figure 4.11.: Comparison of the computed and analytic wave amplitude at the center (10,10) m of the tank for the 3D sloshing problem. Eq. (4.40) defines the analytic solution

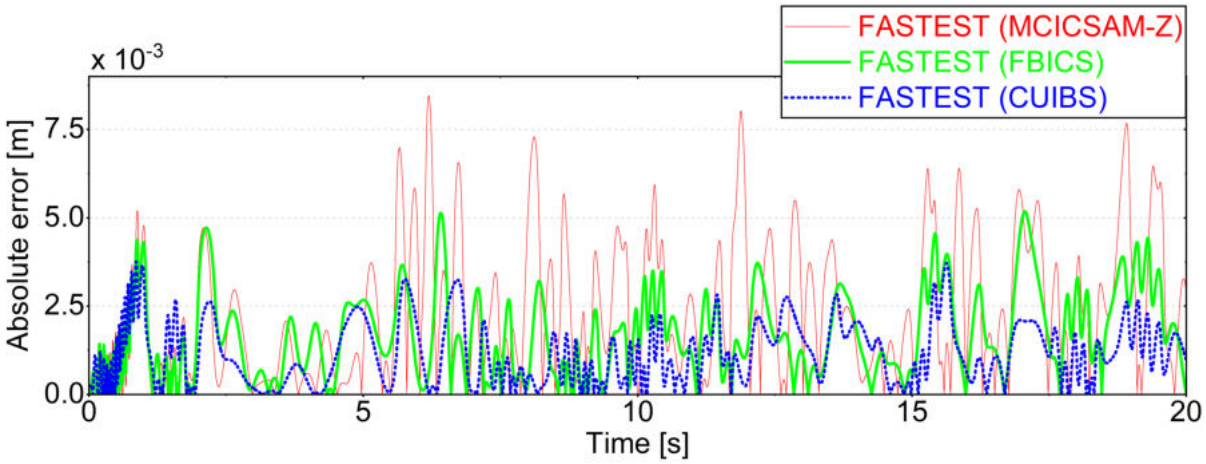


Figure 4.12.: Absolute error of the wave amplitude

Figure 4.11 contrasts the computed wave amplitude's time history at the center of the tank (10,10) m with the analytic solution calculated with Eq. (4.40) during 20 s. We can see an excellent matching between the analytic and computed solution, especially for the curves corresponding to FBICS and CUIBS interface capturing schemes. The maximum absolute error between the computed solutions and the analytical one is about 5×10^{-3} m for FBICS and CUIBS and 8×10^{-3} for MCICSAM-Z, as is shown in Figure 4.12.

Additionally, a time convergence study is carried out to verify the correctness of the selected time-step regarding efficiency and accuracy. Consequently, a higher and a lower time-step are tested. These computation results are shown in Figure 4.13. One can observe that a time-step of 0.005 s introduces instability in the solution. It may be due to the over/undershoots introduced by the BDF2 scheme when large time-steps are used, or the inaccuracy introduced by the interface capture scheme works at high Co. In contrast, 0.0005 s is close to 0.001 s but needs more computational time. So, the selection of 0.001 s was appropriate.

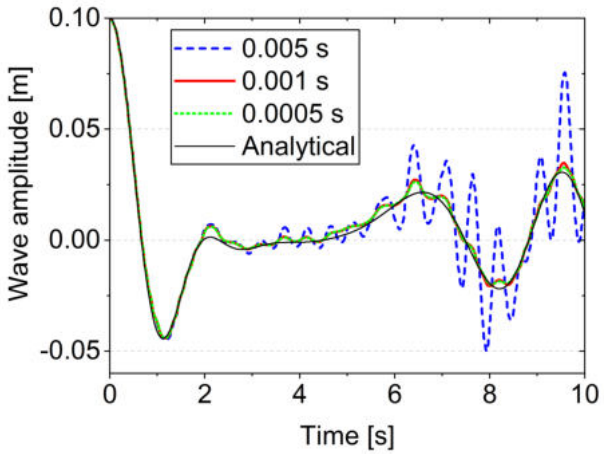


Figure 4.13.: Numerical results of the wave amplitude computed with the FBICS scheme with different time-steps

4.10.2. 3D Rising bubble

The 3D Rising bubble is an example of bubble dynamics with strong surface tension effects. It is a numerical benchmark proposed by Turek et al. in [208]. The benchmarking results were obtained with the FEM-based software FeatFlow [88] and contrasted with the results published by Adelsberger et al. [1]. They used the FD-based solver NaSt3D and the FVM-based software OpenFOAM to compute the problem. This test case validates the two-fluid flow solver implemented in FASTEST, comparing its accuracy with the other mentioned CFD programs.

In a tank full of fluid one, a bubble of fluid two is centered, which rises by gravity effects. The initial geometric configuration of the problem is illustrated in Figure 4.14. All walls are set as no-slip boundary conditions. The density and dynamic viscosity of fluid 1 are $\rho_1 = 1000$ kg/m^3 and $\mu_1 = 10$ Pa·s, and for fluid 2, $\rho_1 = 100$ kg/m^3 and $\mu_1 = 1$ Pa·s. The gravity force is $g = 0.98$ m/s^2, and the surface tension coefficient is $\sigma = 24.5$ N/m.

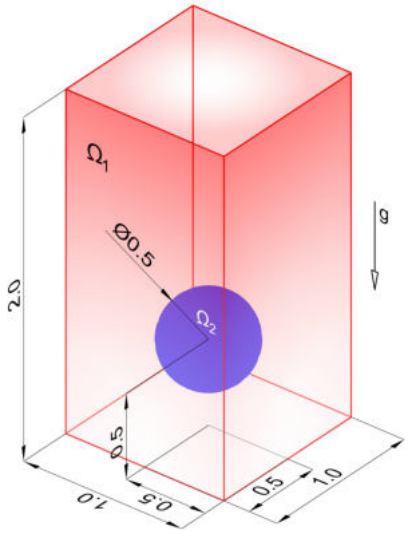


Figure 4.14.: Geometric initial configuration for the 3D rising bubble test case

The selected quantities to compare are the bubble size in two different directions and the rise velocity. The size in the rise direction (z) is determined as R_z/R_0 , and at the perpendicular to the rise direction aligned with the x-axis is R_x/R_0 , where R_0 is the initial bubble size. The rise velocity is defined as

$$v_c = \frac{\sum\limits_{i \in \Omega_2} v_i \delta V_i}{\sum\limits_{i \in \Omega_2} \delta V_i} \tag{4.43}$$

Let v_i be the velocity component in the y-direction and δV_i the size of each CV inside the bubble domain Ω_2.

The computational domain is a uniform structured grid of $80 \times 80 \times 160$ hexahedral control volumes divided into 50 blocks. The BDF2 scheme temporally discretizes the equations, and the time-step is 0.005 s, which yields a maximum local $Co \sim 0.31$ at 0.75 s. The standard sequence (loop-a) of the solution algorithm is employed. First, the volume fraction field is approximated with the MCICSAM-Z, FBICS, or CUIBS capture scheme in a separate iterative process of a maximum of ten iterations and a convergence criterion of $\varepsilon = 5$ times 10^{-4}. Second, the physical properties are updated, and the filtered finite differencing scheme detailed in [63] determines the bubble's curvature. Third, the pressure-velocity coupling algorithm set in PISO mode executes a maximum of 100 iterations per time-step to achieve convergence ($\varepsilon = 10^{-6}$). Under-relaxation is not used for all the variables.

Figure 4.15 contrast the results obtained with the two-fluid solver implemented in FASTEST with the reference results presented by Turek et al. in [208]. The a) rise velocity of the bubble during 3 s, and b) its size in the z and x-directions in the middle vertical plane are close to the reference, especially with the reference results computed with OpenFOAM. Only the size in the rise direction R_z slightly differs from it. This quantity is approximately 0.0125 mm smaller and coincides with the Δz of the CVs. We can observe this small difference in Figure 4.16, where the computed bubble shape with the FASTEST two-fluid solver is compared with the reference

bubble shape obtained with FeatFlow at time 3 s. Furthermore, Table 4.2 groups the monitored quantities of the benchmark test case at time $t = 1$ s and at final time $t = 3$ s.

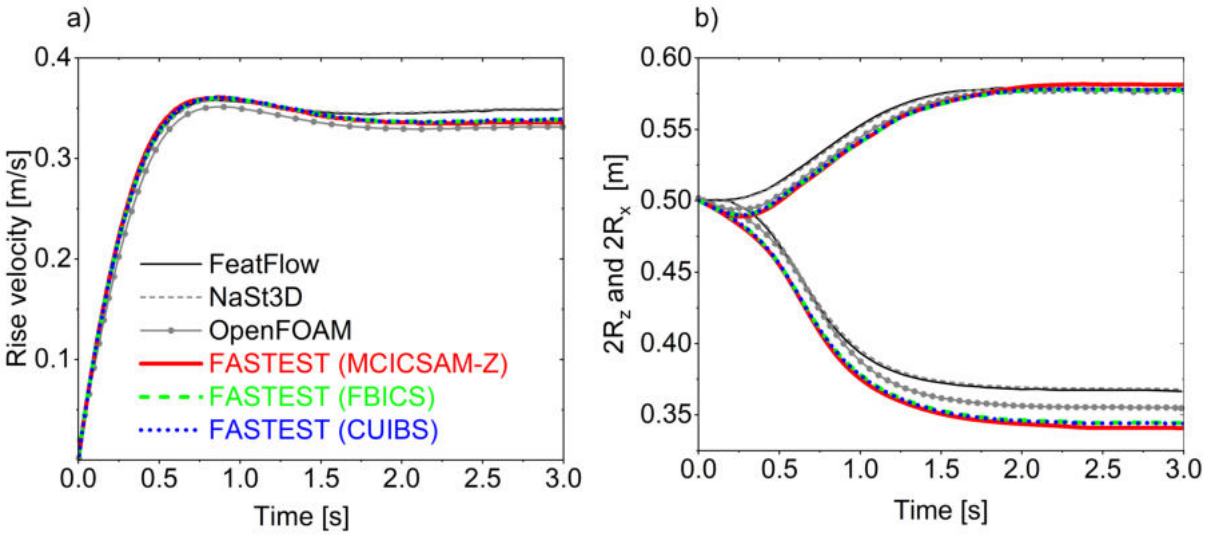


Figure 4.15.: a) Temporal evolution of the rise velocity of the center of the bubble and b) bubble diameter $2R_x$ and $2R_z$ computed with the two-fluid solver implemented in FASTEST. The MCICSAM-Z, FBICS, and CUIBS blended schemes are used for the calculation. NaSt3D, OpenFOAM, and FeatFlow refer to the reference results presented in [208]

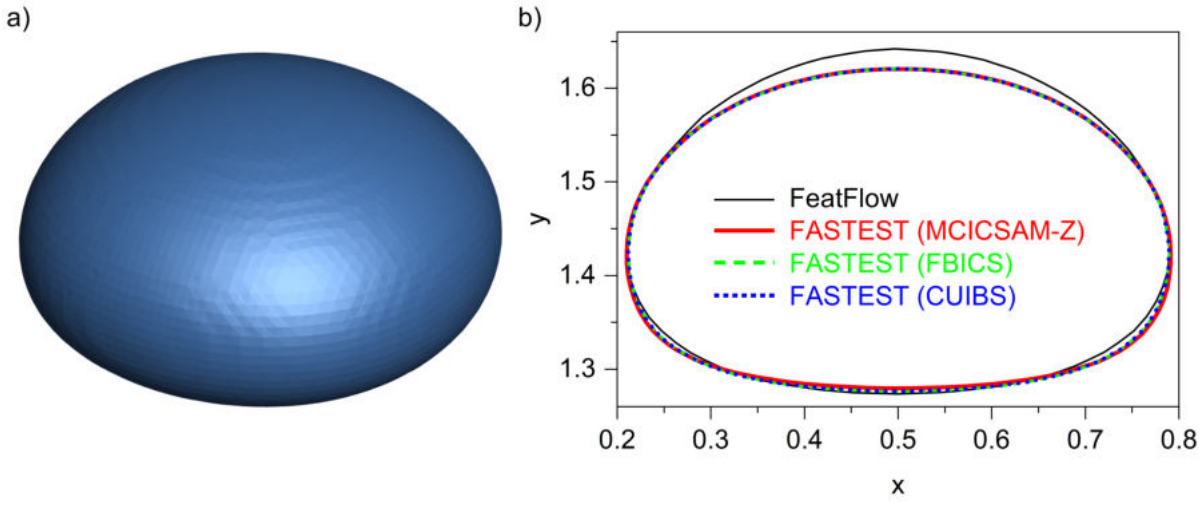


Figure 4.16.: a) 3D Bubble shape at 3s obtained with FASTEST and using the CUIBS scheme to advect the volume fraction, and b) bubble shape in the vertical middle plane contrast with the benchmark results computed with FeatFlow

I assume that the minor difference clustered in the bubble's vertical size is related to the surface tension model since this is the dominant effect in this test case. Besides, testing the different

alternatives available in the FASTEST code for the surface tension calculation (see [189] for more information about the methods), I observed that the shape changes drastically according to the method used. However, the surface tension calculation is not the focus of this work.

Solver	$t = 1.0$ **s**				$t = 3.0$ **s**			
	zc	**vc**	**Rz/R0**	**Rx/R0**	**zc**	**vc**	**Rz/R0**	**Rx/R0**
FeatFlow	**0.276**	**0.357**	**0.787**	**1.106**	**0.973**	**0.349**	**0.734**	**1.158**
FASTEST:								
MCICSAM-Z	0.273	0.358	0.750	1.082	0.944	0.336	0.682	1.163
FBICS	0.279	0.359	0.756	1.082	0.943	0.338	0.689	1.154
FASTEST	0.279	0.359	0.755	1.083	0.943	0.339	0.688	1.156

Table 4.2.: Monitored quantities at $t = 1$ s and at final time $t = 3$ s. The symbol z_c is the z location of the bubble associated with its point of mass, v_c is the rise velocity, and R_z/R_0 and R_x/R_0 are the bubble's size parameters

Regarding the computational effort, the pressure-velocity coupling algorithm needs an average of 11 iterations per time-step regardless of the interface capturing scheme used. While the number of iterations necessary to solve the VoF equations drastically differs, MCICSAM-Z needed around nine iterations per time-step to achieve convergence, FBICS 6 iterations, and CUIBS 4 iterations.

4.10.3. Rayleigh–Taylor instability

If a heavy fluid of density ρ_2 is superimposed on a lighter fluid of density ρ_1 in a vertical gravity field g, the fluid interface is unstable. The interface's perturbation grows with time [31], and the interface rolls up, which results in a mushroom pattern [167]. This phenomenon is known as Rayleigh-Taylor instability, and it is used to test the capability of the method for capturing wave breaking.

The solution algorithm solves the two-fluid flow case with the same kinematic viscosity ($\nu_1 = \nu_2$). The numerical results are compared with the analytical results obtained by Chandrasekhar [25]. He predicts a specific variation of the linear growth rate with the Reynolds number for many density ratios. Besides, he shows that the early phase of unstable growth is a function of density ratio, gravitational acceleration, and kinematic viscosity. Thus, the variation of growth rate is given by $n = g^{2/3}\nu^{-1/3}$ (inverse time units), the wavenumber by $k = g^{1/3}\nu^{-2/3}$ (inverse distance units), and the Reynolds number by $Re = \lambda^{3/2}g^{1/2}\nu^{-1}$. The purpose of this section is to compare the results using MCICSAM-Z, FBICS, and CUIBS with his prediction for the specific density ratio 2:1.

The test settings follow the recommendations of references [31, 167]. The computational domain is a 2D rectangle defined by the opposite vertices (0, -0.03) m and (0.02, 0.03) m and discretized with a uniform grid of 40×120 CVs. The four boundaries are treated as slip boundaries. The lighter fluid density is $\rho_1 = 1$ kg/m^3, the gravitational acceleration $g = 1$ m/s in the y-direction. Initially, the interface is located at $y = 0$, and a single wavelength perturbation is

introduced through the following velocity field

$$u = \frac{\pi A \delta y}{2L} \left(2H\left(y\right) - 1\right) \sin\left(\frac{\pi x}{L}\right) \exp\left(\frac{-\pi \left|y\right|}{L}\right),$$
$$v = \frac{\pi A \delta y}{2L} \cos\left(\frac{\pi x}{L}\right) \exp\left(\frac{-\pi \left|y\right|}{L}\right), \tag{4.44}$$

where $A = 0.1$ and represents the amplitude of the perturbation, $L = \lambda/2$ is the half wavelength of the perturbation and corresponds to the width of the mesh (0.02 m), δy is the mesh spacing in the y-direction, and $H(y)$ is the Heaviside function given by

$$H\left(y\right) = \begin{cases} 1 & y > 0 \\ 0 & y \leq 0 \end{cases} \tag{4.45}$$

The numerical results are obtained for the Reynolds numbers 39, 72, and 176 using the three interface capturing schemes. Figure 4.18 shows the interface's position at four different times approximated by the CUIBS scheme for the above Reynolds numbers. The instability increases as the Re increases; however, the interface remains symmetric for the three conditions until 0.5 s. Then, the mushroom pattern starts to appear, as we can see at time 1.5 s for the middle Re.

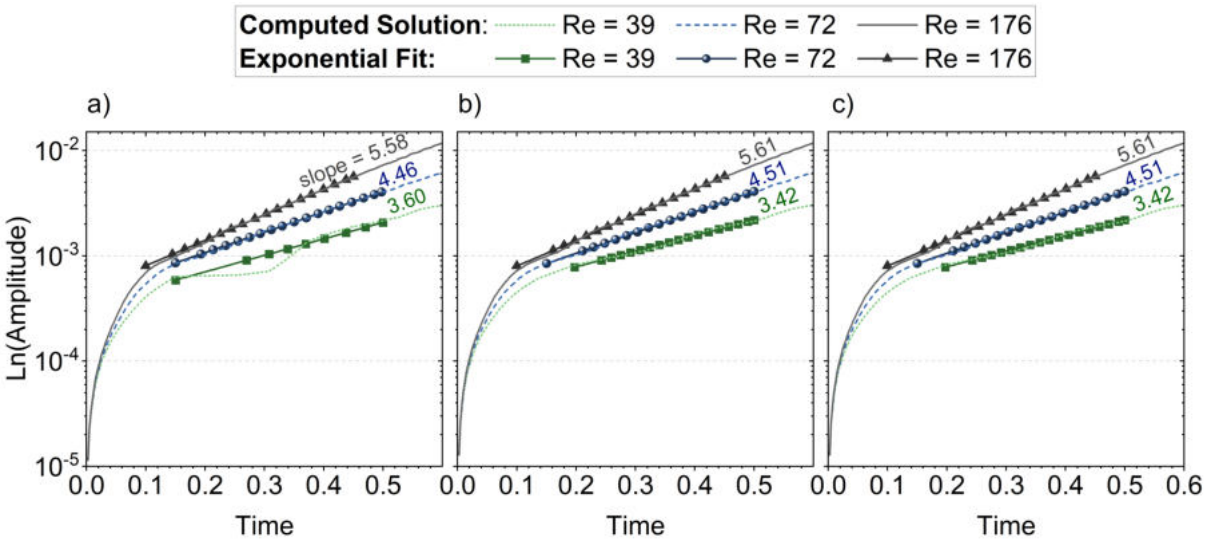


Figure 4.17.: Estimation of the linear growth rate of Rayleigh-Taylor instability using a) MCICSAM-Z, b) FBICS, and c) CUIBS

Figure 4.17 depicts the logarithm's temporal evolution of the amplitudes for the computation with a) M-CICSAM, b) FBICS, and c) CUIBS of the studied Reynolds numbers. The amplitude is the average of the interface's absolute displacements along the left and right vertical walls. The slope of these amplitude curves measures the exponential growth rate n of the instabilities. An exponential fit of only the linear region of the curve determines the slope to avoid the transient modes of the early phase of instability. For $Re = 39$ and 72 this region is [0.15 - 0.5] s, while for $Re = 176$ is [0.1 - 0.45] s. The growth rate obtained by a) MCICSAM-Z differs slightly from

those computed with b) FBICS and c) CUIBS, which coincide for the three Reynolds numbers. These values are 3.42, 4.51, and 5.61 for $Re = 39$, 72, 176, respectively, and were close to the values obtained first by Daly [31]: 3.6, 4.5 and 5.6, and later by Queutey [167]: 3.5, 4.7, and 5.5 for the same Reynolds numbers. The differences in the results calculated using MCICSAM-Z, especially for the most viscous flow ($Re = 39$), are presumed to be related to the scheme itself. It tends to align the interface with the grid near the boundaries lightly. Maybe this numerical perturbation affects the linear flow condition necessary for the numerical calculation's success.

Table 4.3 presents an analysis of the three schemes from the point of view of computational efficiency. For all the calculations, the enhanced pressure-velocity coupling algorithm in mode PISO is used, and the maximum number of iterations is set to 100. An iterative process of a maximum of ten iterations solves the VoF equation before beginning the pressure-velocity calculation. No under-relaxation is used for all the variables, and the convergence criterion for the velocity and pressure fields is 10^{-6} and for the volume fraction 10^{-5}. The time-step is constant and equal to 0.0025 s for all the scenarios resulting in maximum Courant numbers of 0.28, 0.68, and 0.98 for the Re number analyzed. The reader can see that the computations of velocity and pressure field for the Rayleigh-Taylor instabilities require the same average of PISO iterations independent of the method used for solving the VoF equation.

Re	**max Co**	**PISO Iterations**			**VoF iterations**		
		MCICSAM-Z	FBICS	CUIBS	MCICSAM-Z	FBICS	CUIBS
39	0.28	24	24	24	8	6	4
72	0.68	13	13	13	9	7	6
176	0.98	16	16	16	9	8	7

Table 4.3.: Average number of SIMPLE step iterations and VoF iterations per time-step to solve the Rayleigh-Taylor instability for different Reynolds number

In contrast, the VoF iterative process becomes significantly different depending on the interface capturing scheme employed for the interface approximation. At low Co conditions, CUIBS solves the problem two times faster than MCICSAM-Z and 50% faster than FBICS. Likewise, CUIBS is the fastest, followed by FBICS and MCICSAM-Z for the medium and high Co numbers. MCICSAM-Z has the worst performance because it does not achieve the convergence criterion in a big part of the calculations. Moreover, the three schemes have numerical difficulties that delay their convergence even though they define sharp interfaces at high Co numbers.

Finally, using the most efficient advection scheme, CUIBS, the computations for $Re = 10$, 100, 200, 300, 400, and 500 are performed. The amplitude growth slopes are again determined in the region of [0.15 - 0.5] s for $Re = 10$, and [0.1 - 0.45] s for the other. These growth rates n are plotted in dimensionless form against Reynolds number in Figure 4.19, and they are contrasted with the analytical prediction of Chandrasekhar. An excellent overall agreement with the theoretical curve is observed.

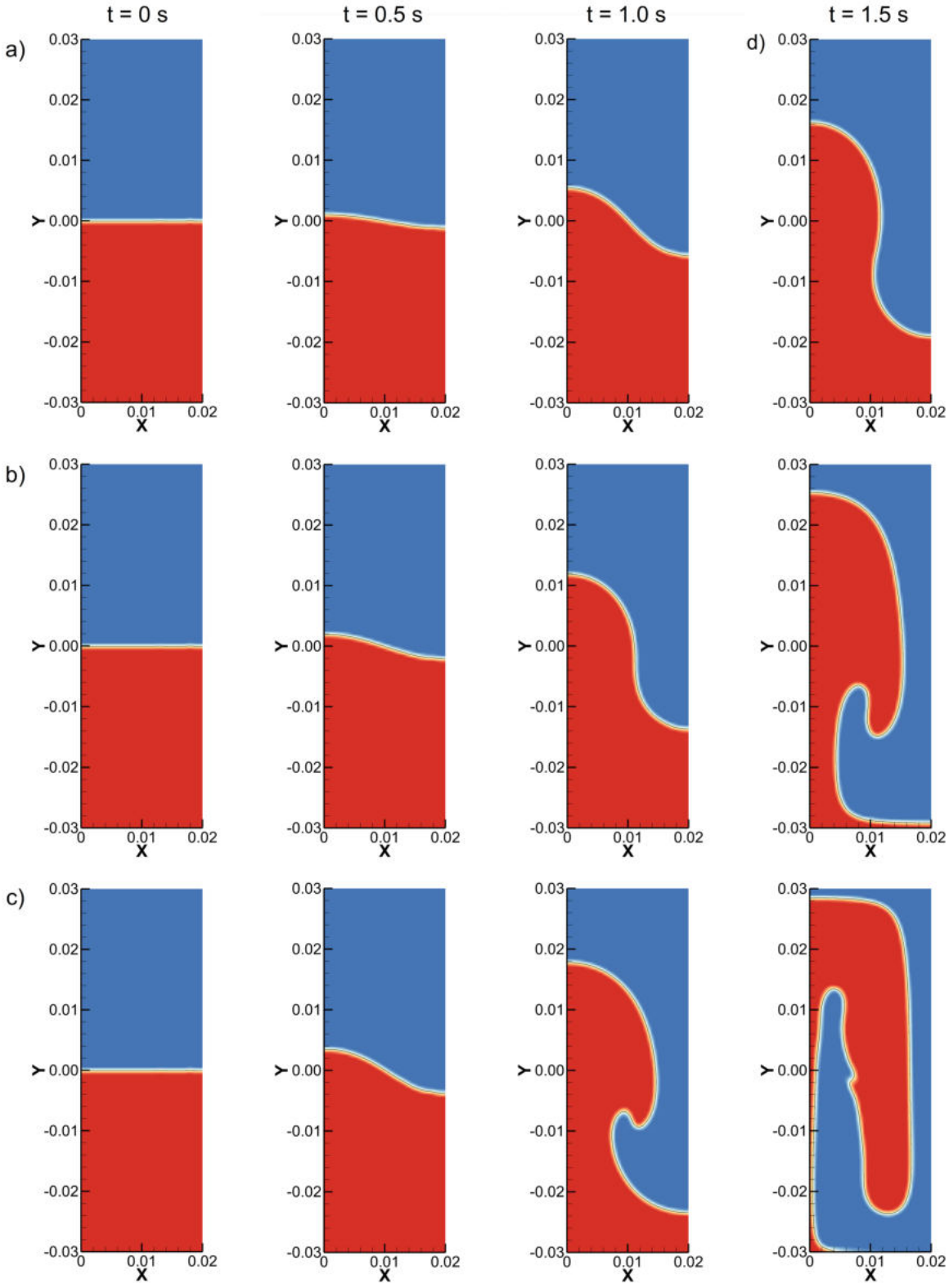


Figure 4.18.: Snapshots at different times of the Rayleigh-Taylor instability computed with the CUIBS interface capturing scheme at a) $Re = 39$, b) $Re = 76$, and c) $Re = 176$

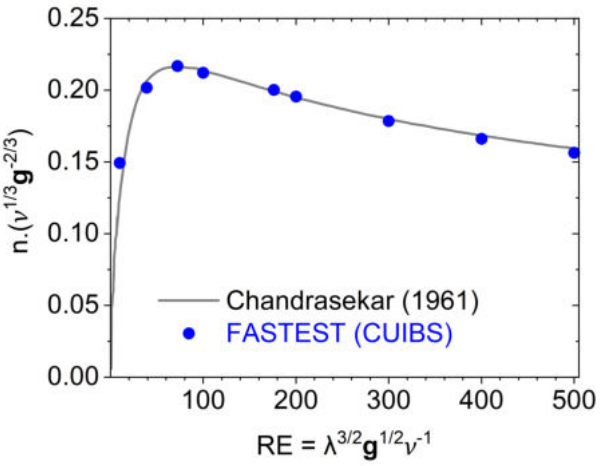


Figure 4.19.: Non-dimensional perturbation growth rate versus Reynolds number for Rayleigh-Taylor instability computed the FASTEST solver and the analytic solution of Chandrasekar

4.10.4. The dam break flow impacting a rigid structure

In this section, to validate the two-fluid flow solver for more practical applications, the classic dam breaking example is computed. It was experimentally studied by Koshizuka [108] to describe the collapse of a water column impacting a rigid structure.

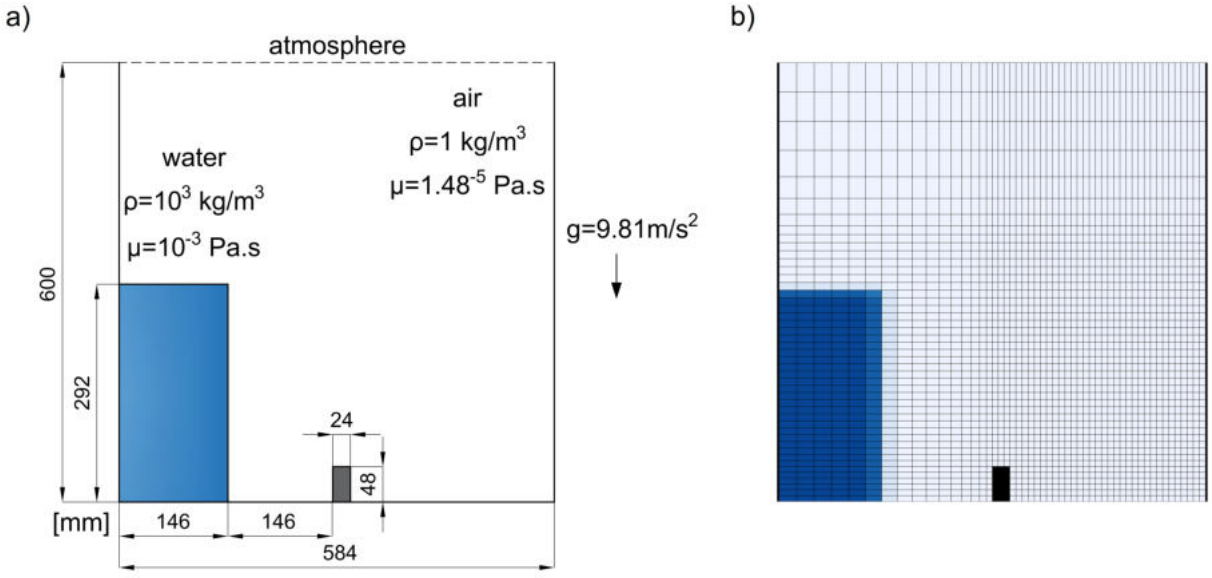


Figure 4.20.: a) Geometrical configuration and physical parameters of the dam break flow impacting a rigid structure test case. b) Initial condition in the coarse computational domain

The problem's geometry and physical parameters are depicted in Figure 4.20. It consists of a tank open to the atmosphere that contains a water column that collapses and hits a rigid obstacle. The high-density fluid is water, and the low-density fluid is air, both with standard physical properties (as seen in Figure 4.20a). The two-phase flow is considered laminar, and the surface tension effects are neglected.

The computation runs for three grid levels consisting of 2328 CVs, 9312 CVs, and 37 284 CVs. The meshes have a biexponential distribution, as is shown in Figure 4.20b. The automatic time-step refinement is chosen for these computations because the water's kinematic energy increases with time, which produces a continuous increase of the Courant number. If a small constant time-step is selected, the calculation is stable but inefficient, while a large value leads to divergence when the water's velocity is higher.

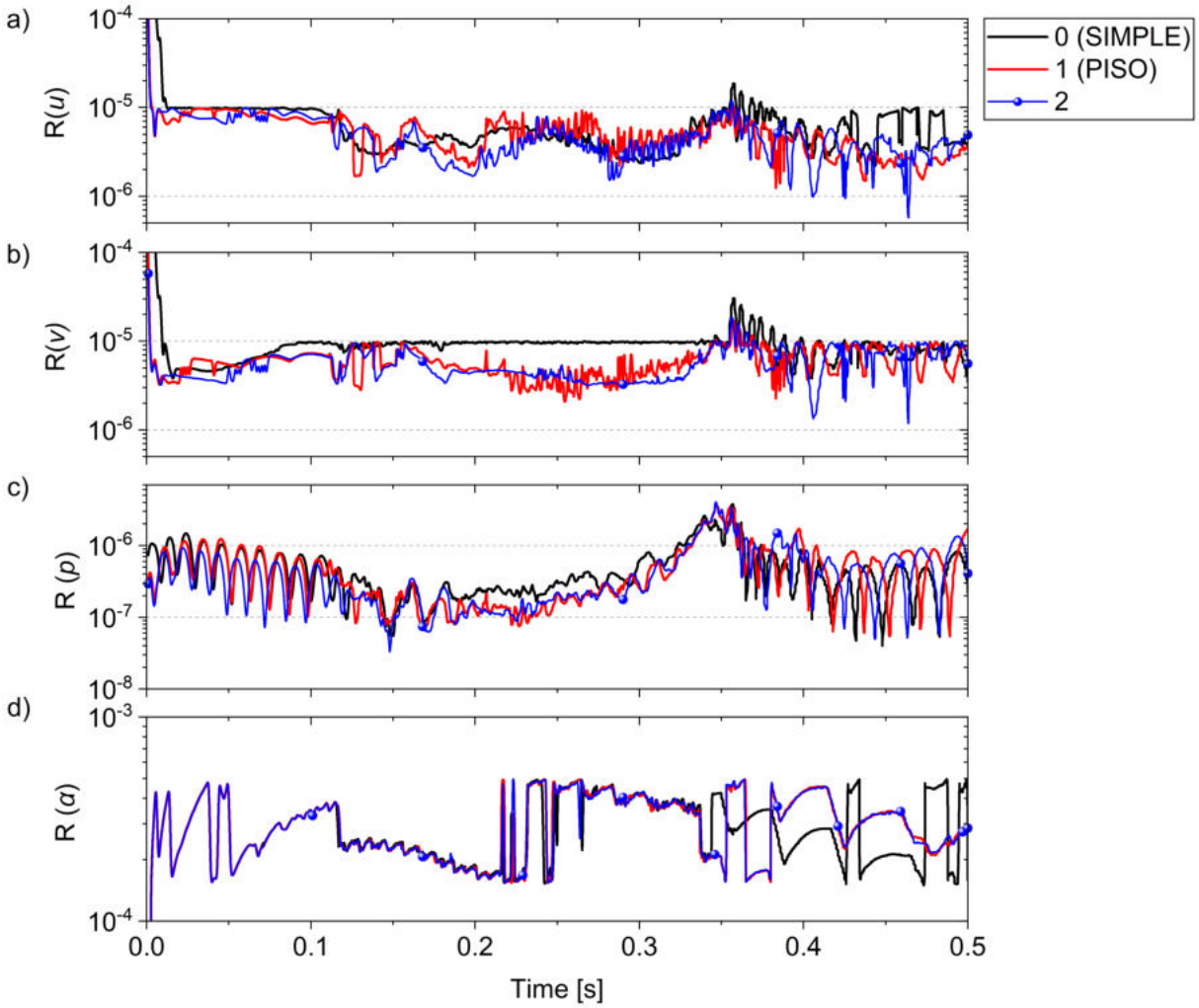


Figure 4.21.: Evolution in time of the normalized numerical residuum for the velocity components, pressure, and volume fraction fields solving the dam break flow, impacting a rigid structure test case for the coarse grid

Firstly, a parametric study is carried out using the coarse grid to find more appropriate settings for the case. For this preliminary study, the CUIBS scheme advects the volume fraction, while the QUICK scheme approximates the momentum equation's convective term. The iterative solution process of the VoF equation is limited to 20 iterations, and the convergence parameter is 5×10^{-4}. The pressure-velocity coupling algorithm is set to perform a maximum of 50 iterations with *zero, one,* and *two* PRIME steps. The solution converges when the normalized numerical errors are less than 10^{-5}. The under-relaxation factor for the velocity field is $\gamma^{\mathbf{v}} = 0.7$, for the pressure filed $\gamma^{p} = 0.3$, and for the volume fraction field $\gamma^{\alpha} = 0.7$. The automatic time-step refinement is activated to maintain the local Co in the range of 0.5 to 0.8. A summary of the computational effort required by each setting for solving the first 0.5 s of the dam breaking phenomenon is presented in Table 4.4, while the evolution of the dependent variable's numerical residues is depicted in Figure 4.21. As demonstrated with the one fluid case in Section 3.8.2, the proposed enhanced SIMPLE algorithm's performance increases when more PRIME steps are performed. However, the extra calculations consume time. Therefore, the optimal setting seems to be the PISO mode. It solves the problem with 40% less computation time than SIMPLE. Moreover, the number of iterations that solve the VoF equation is in a similar range for the three options, although in the last period ([0.4 - 0.5] s), the α-residues are higher for SIMPLE.

PRIME steps	Pressure-velocity iterations	VoF iterations	Real time [s]
0 (SIMPLE)	16777	2932	95.8
1 (PISO)	4756	2628	57.5
2	4105	2621	66.1

Table 4.4.: Total number of pressure-velocity coupling iterations, total VoF iterations, and the total real computational time required to simulate 0.5 s of the dam break flow impacting a rigid structure case using zero, one, and two PRIME steps

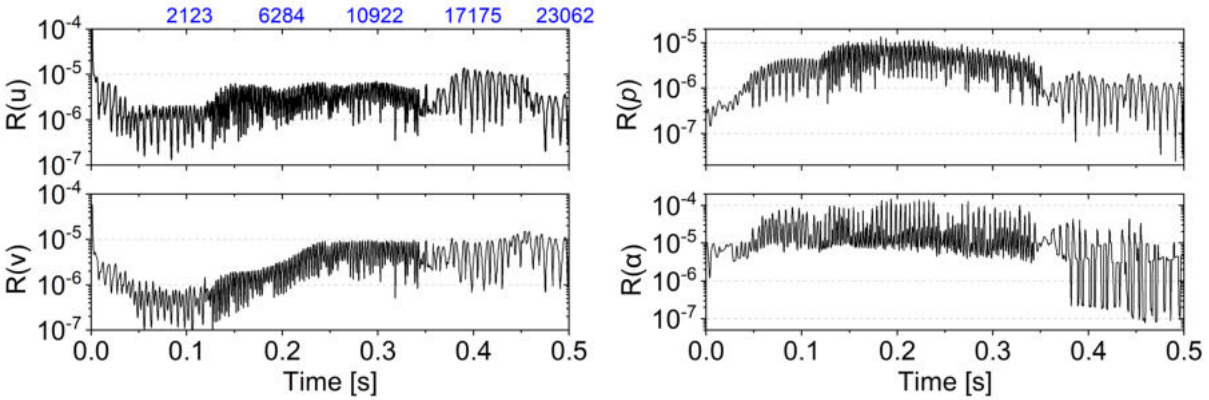


Figure 4.22.: Evolution in time of the normalized residues for the dependent variables solving the dam break flow impacting a rigid structure test case for the coarse grid when the VoF equation is included in the pressure-velocity loop. The blue numbers correspond to the number of iterations performed in each time

The loop-b of the solution algorithm, including the VoF equation in the pressure-velocity coupling loop, is contrasted with the results calculated above using *two* PRIME steps for which the loop-a was applied, solving the VoF equation in a separate loop. Figure 4.22 shows the evolution of the residues applying loop-b. Even a better numerical accuracy is achieved for the velocity and volume fraction field. This alternative is enormously time-consuming. It requires 23062 iterations of all variables that take 437 s. Thus, loop-b needs 5.6 times more iterations and requires about seven times more computing time than loop-a. According to the pressure residues' evolution, we can interpret that the constant change in physical properties increases the non-linearities in the pressure-correction equation, which decreases its convergence rate.

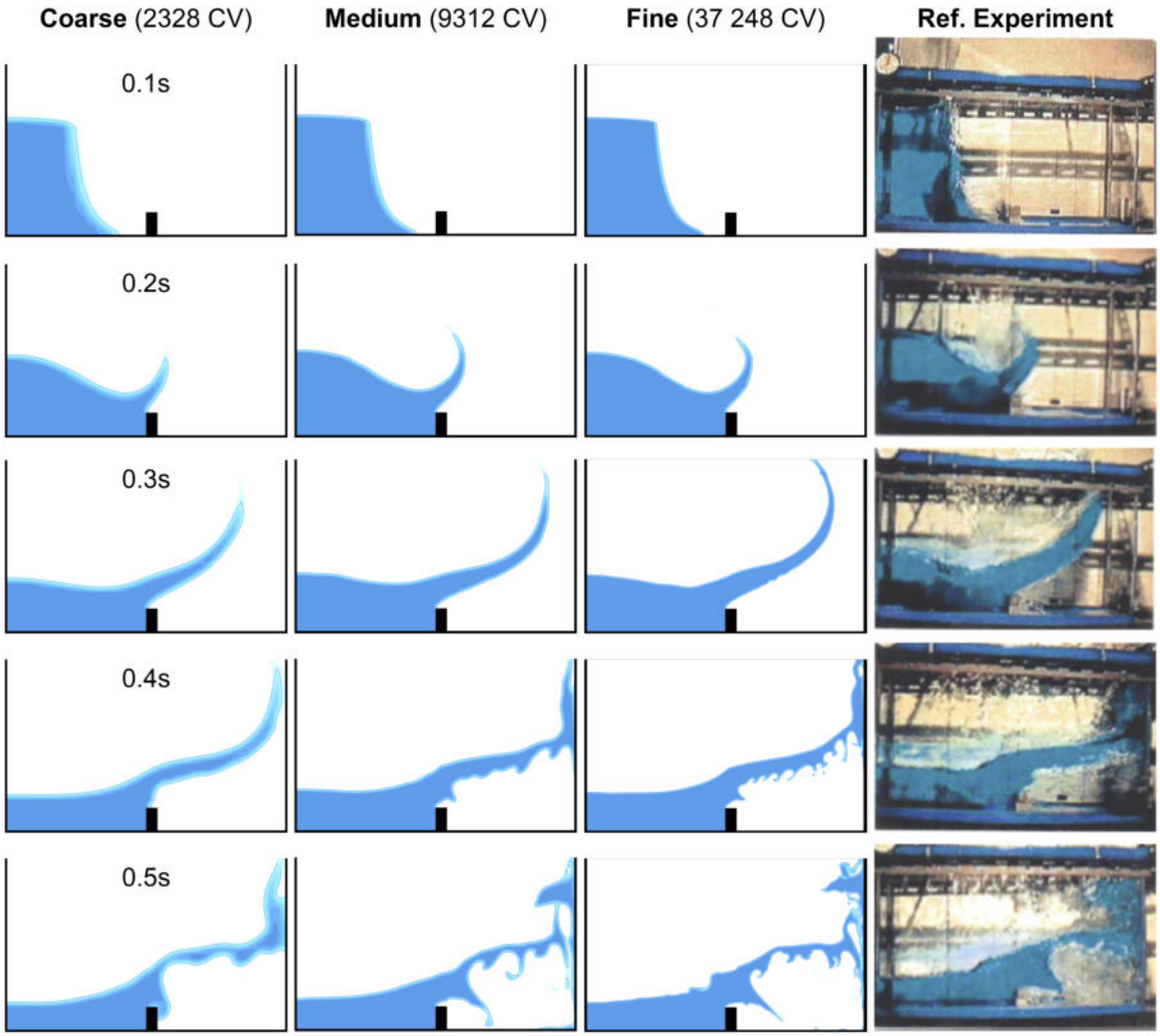


Figure 4.23.: A comparison of the free surface profile in the dam break experiment of Koshizuka (1995) at the same snapshot times. The free surface is capture with the CUIBS interface capturing scheme

After the parametric study, the problem is solved for the coarse, medium, and fine grid utilizing the best settings, loop-a of the solution algorithm, and the PISO mode. The convergence criterion for the VoF equation is increased to 10^{-5} and for the velocity and pressure fields to 10^{-7}. The time-step refinement controls that the Co number stays in the range of 0.6 to 0.8. Table 4.5 collects the efficiency quantities for the three interface capturing schemes of interest: MCICSAM-Z, FBICS, and CUIBS to solving the dam break flow impacting a rigid structure test. The three schemes result in a similar efficiency at the three grid levels. However, MCICSAM-Z diverges at 0.3 s of the computation for the medium level, and the CUIBS scheme is slightly faster at the fine grid level. The numerical results are similar for the three schemes. Thus, only the results computed with the CUIBS scheme are contrasted with the experiment of Koshizuka [108] in Figure 4.23, which are in excellent agreement as the mesh is refined.

Scheme	**Coarse**			**Medium**			**Fine**		
	Iterations		Time	Iterations		Time	Iterations		Time
	PISO	VoF	[s]	PISO	VoF	[s]	PISO	VoF	[s]
MC.-Z	29036	7342	297	*	*	*	81222	64816	5570
FBICS	31100	7310	314	41227	23122	1210	75871	67424	5280
CUIBS	31199	6206	321	38860	21199	1280	67507	51268	4660

Table 4.5.: Total number of PISO iterations, VoF iterations, and time required by each interface capturing scheme for solving the dam break flow impacting a rigid structure case using three grid levels of refinement. * detones divergence

Conclusions:

- The two-fluid flow solver's accuracy and efficiency have been validated through the above four test cases. The obtained solutions are in good agreement with analytical, numerical, and experimental reference solutions.
- The best interface capturing schemes for working at medium and high Co conditions combined with the two-fluid solver are CUIBS and FBICS. They present low numerical diffusion and allow using a large time-step without a loss of stability. Therefore, they are the best candidates to be used for the next step, integrating the two-fluid solver into a partitioned approach to simulate free surface flows in interaction with flexible structures.

5. The fluid-structure interaction coupling

The growing accuracy and efficiency requirements in many naval and civil structures' design tasks increasingly imply the need to consider the interaction between the surrounding flow and the elastic structure. This engineering requirement calls for multiphysics simulations that precisely solve the interaction between free surface flows and deformable elastic structures, i.e., fluid-structure interaction (FSI) computations, including two-fluid flows solvers and elastic structure models. Although several numerical solution frameworks have been published in recent years, often with good results, some key points have not yet been satisfactorily resolved. One of them is the necessity of reducing the long computational time typical for these simulation types. It makes its application in the design stage unpractical, as it often needs to be finished in a couple of weeks. Another point is the design of robust and efficient solvers that can address high nonlinearities, e.g., substantial structure deformation and violent free surface flows, without a lack of stability, and the development of more experiments that validate the numerical models and measure the reliability of the results.

This chapter focuses on redesigning an efficient and robust numerical partitioned FSI approach that includes free surface flows in its solution framework. The new FSI methodology is a redesign of the approach initially developed by Schäfer, and Teschauer [187]. Therefore, Section 5.1 begins the chapter by giving a general background of the original approach and points out its issues and the motivation for the changes. Section 5.2 describes the new redesigned FSI coupling algorithm, based on the fluid solver FASTEST, the structural solver CalculiX, and the multiphysics coupling library preCICE. The most important details about the communication and data mapping, the acceleration techniques, and the coupling configuration are detailed. Subsequently, in Section 5.3 the attention is given to showing the implementations in the FASTEST code to be coupled in the multifield framework, i.e., how is the coupling Adapter structured and how the calculation of the two-fluid flows is included. The calculation of the grid velocity introduced by the ALE description is given in Section 5.4. Finally, the new redesigned FSI coupling algorithm is validated in Section 5.5 and used to solve three FSI test cases involving free surface flows in Section 5.6.

5.1. Background of the FSI partitioned approach

Originally, FASTEST was part of an implicit partitioned fluid-structure interaction approach in the ALE framework developed by Schäfer and Teschauer [187]. This approach coupled FASTEST as fluid solver with the finite-element structural solver FEAP [200] through the commercial quasi-standard coupling interface MpCCI [183]. The solution procedure consisted of a predictor-corrector iterative sequential procedure controlled within FASTEST and accelerated by multigrid methods, the adaptative under-relaxation of Aitken, and displacement predictions [180, 184, 191, 240]. The fluid mesh deformation employed algebraic linear and transfinite interpolations and an

elliptic technique [186, 241]. The MpCCI coupling interface controlled the data communication and the mapping between the non-matching fluid and structure grids.

This FSI coupling approach was successfully applied to solve several one-fluid FSI problems, see, e.g., [73]. However, unfortunately, due to the updates of the application programming interface (API) of MpCCI, the solvers had compatibility issues that were not easy to repair and drove to the exchange of MpCCI for the Precise Code Interaction Coupling Environment (preCICE) [19]. The coupling library preCICE is open-source software that offers different communication means, data mapping schemes, and iterative methods for solving the fixed-point equations. The preCICE API operates on a generic level, allowing easy and fast implementation in in-house codes.

The first attempts to rebuild the FSI coupling entailed replacing each MpCCI API with its equivalent preCICE API ignoring that MpCCI is a mixture between library and framework approach, and preCICE a pure library approach. Hence, its capabilities differ and cannot be used in the same way [212]. As a result, the partitioned FSI coupling between FASTEST and FEAP via preCICE led to an inconsistent FSI coupling. Finding the solution for the coupling troubles was a big part of this research work and motivated to redesign the FSI coupling algorithm.

After various analyses of the troubles, the coupling failure causes were identified as the improper use of the preCICE APIs and the ”Adapters” incorrect structure. An Adapter is a piece of code written in the Solver Language that introduces the coupling library’s APIs into the program and enables the program to receive and send information. The solution required to rebuild the fluid solver adapter and the structure solver adapter almost from scratch.

Building an Adapter can be an easy or difficult task depending on the software developer’s experience with the program because the APIs need to be included in a specific part. Thus, in-depth knowledge of the program structure is mandatory. This level of expertise we had with the FASTEST, while the structure of the program FEAP was a black box. So, I rebuilt the Adapter for the FASTEST solver and decided for convenience to replace the structural program FEAP with the open-source finite element program CalculiX [43] because this offers an official ready-to-use preCICE adapter [213].

5.2. The Redesigned partitioned FSI coupling approach

This new partitioned FSI coupling approach couples the finite volume fluid solver FASTEST with the finite element structural solver CalculiX using the multiphysics library preCICE. The iterative coupling procedure is schematized in Figure 5.1. After the Solvers’ initialization, at the beginning of each time-step, FASTEST and CalculiX save the old state of the variables and geometry. The fluid solver reads from the interface the last available displacements, transforms them into velocities, and sets them as Dirichlet boundary conditions. Subsequently, FASTEST updates the fluid grid and solves the flow field in the current flow geometry. From this, the fluid-induced forces at the interface are calculated. These forces are set into the structural solver as Neuman boundary conditions and used to computes the structural deformations. preCICE steers the communication between solvers, the data mapping, and the iterative process. When the balance of displacements and forces do not achieve convergence, preCICE forces the Solvers to reset the old state variables and repeat the loop. Then, preCICE allows the solvers to advance in time, and the calculation starts once again.

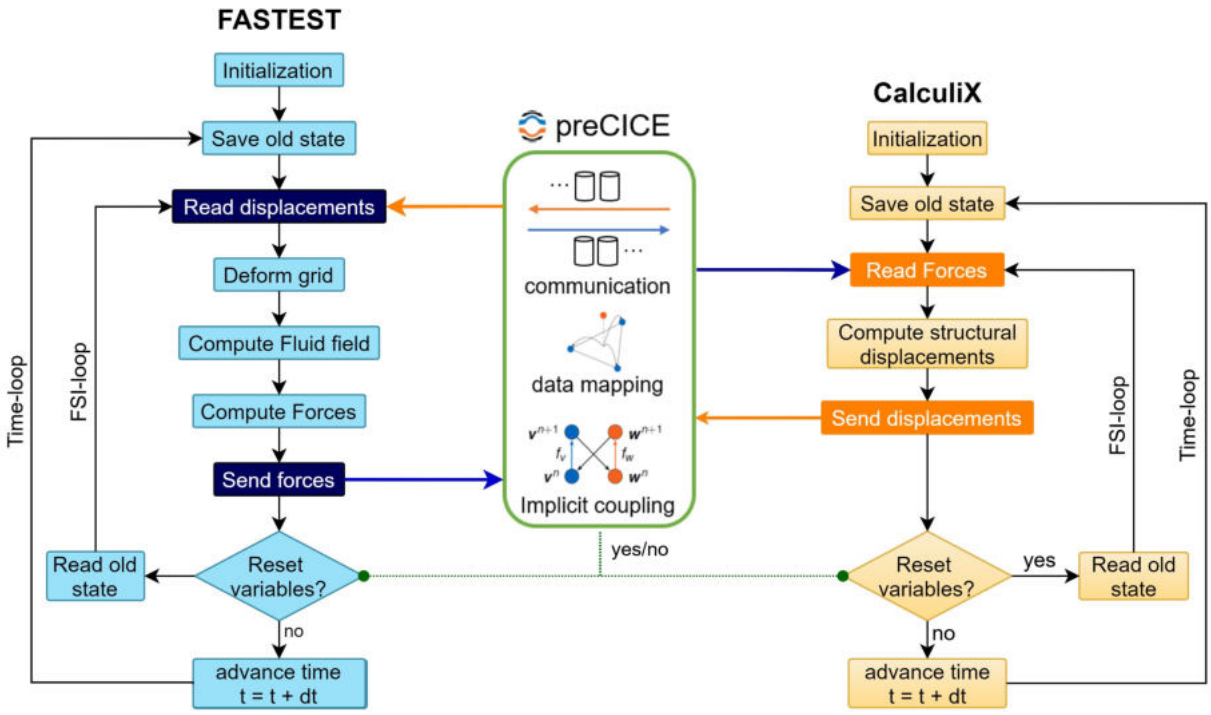


Figure 5.1.: Scheme of the FSI coupling algorithm

5.2.1. Communication and data mapping

The data communication between the coupling surfaces corresponding to the fluid solver and the structural solver is performed via preCICE. The communication is either based on MPI ports or low-level TCP/IP sockets. Figure 5.2 schematizes the information flow. Besides, in the case of non-conforming meshes, preCICE is responsible for data mapping.

After the initialization, the library preCICE is supplied with the coupling meshes that are the surface meshes defined by the structure and fluid solver at the common coupling interface. FASTEST sends two coupling meshes, one given by the coordinates of the CV-vertices and another given by the CV-centers at the interface. The first is named *Fluid_Nodes*, and the second *Fluid_Center*. From CalculiX, only the mesh *Structure_Nodes* formed by the coordinates of the nodes is sent. The meshes of both participants are analyzed by preCICE to construct local communication channels to write and read information. FASTEST uses the *Fluid-center* mesh to write the Surface Forces and the *Fluid_Nodes* mesh to read the displacements. Whereas, CalculiX reads the forces and writes the displacements in the same mesh, the *Structure_Nodes* mesh.

For data mapping, preCICE supports three methods: Nearest Neighbor mapping, Nearest-Projection mapping, and Radial Basis Function mapping. In the first order **Nearest Neighbor** (NN) mapping, a data point at the target mesh is assigned the value at the data point of the source mesh that is the closest to its position. While the second-order **Nearest-Projection** (NP) mapping is based on projections of data points of the target mesh to mesh elements of the source mesh and a (bi-)linear interpolation scheme in these elements [22]. On the other hand, the **Radial Basis Function** (RBF) mapping builds up a global interpolation using radial symmetric

basis functions centered at the source mesh's grid points. For a more detailed introduction to the mapping methods, the reader may refer, for instance, to [67, 212].

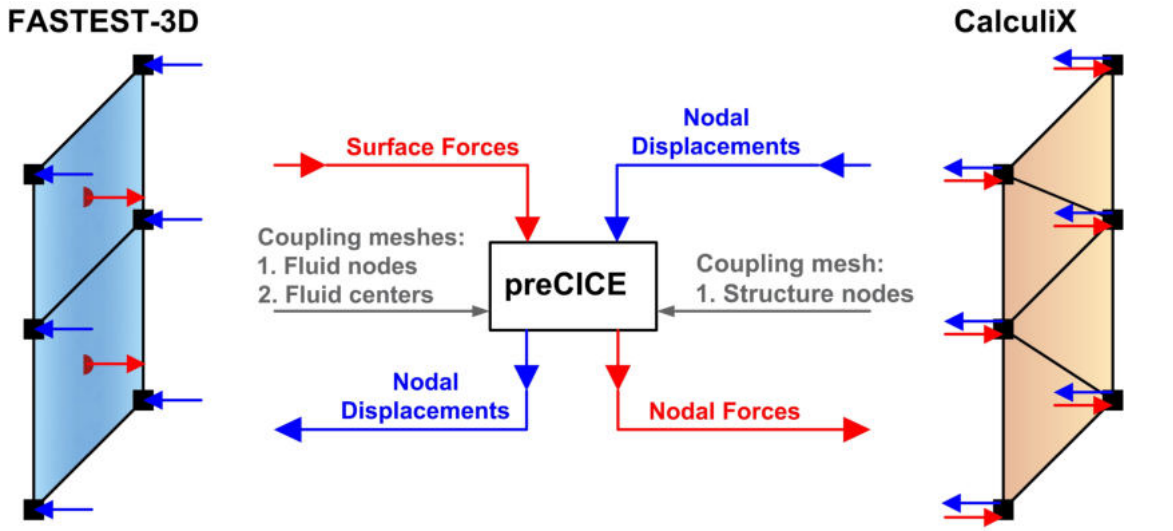


Figure 5.2.: Schematic representation of data flow between fluid and structure solvers via the preCICE coupling library

The three mapping methods can map the couple variables in either conservative or consistent form. The conservative approach is based on the global conservation of virtual work over the interface [51], while the consistent approach performs an exact interpolation over the interface [37]. For example, using a NN mapping from a fine to a coarse grid, the **consistent** approach assures that the value at a coarse node is the same as the corresponding fine node's value. The **conservative** approach computes the value at a coarse node as an aggregation of the corresponding fine nodes [37, 220]. Usually, the consistent condition is applied to displacements transfer, and the conservative condition for forces transfer [212]. According [37], a very efficient coupling strategy uses two different mapping methods to interpolate displacements and forces. For instance, an accurate interpolation method (e.g., RBF) combined with a consistent formulation to interpolate from coarse to the fine grid and a less accurate but faster method (e.g., NN) to interpolate from fine to coarse conservatively.

5.2.2. Iterative solvers for interface coupling conditions

The interface coupling conditions form the fixed-point equations that require either a small and fixed number of solvers calls per time-step or requires an iterative procedure to achieve convergence of the coupling system in each time-step. In preCICE, the first refers to an explicit scheme and the second to an implicit scheme. They can be used depending on the complexity of the multiphysical coupling problem. Since the implicit scheme is more suitable than the explicit one for high-density fluids [21], only this scheme is described below.

We assume the operators F and S are the black-box fluid and the structural solvers, respectively. The displacements d and forces f are the coupling variables. preCICE can manage the fixed-point

iteration process with a serial or parallel execution of the solvers. The serial manner is based on a Gauss-Seidel execution of both solvers:

$$d^{(n+1),i+1} = F^{(n)}\left(f^{(n+1),i}\right) \quad \text{and} \quad f^{(n+1),i+1} = S^{(n)}\left(d^{(n+1),i+1}\right), \tag{5.1}$$

whereas the parallel manner simultaneously executes the solver and computes the fixed-point equations as a vectorial system:

$$\begin{pmatrix} f^{(n+1),i+1} \\ d^{(n+1),i+1} \end{pmatrix} = \begin{pmatrix} 0 & S^{(n)} \\ F^{(n)} & 0 \end{pmatrix} \begin{pmatrix} f^{(n+1),i} \\ d^{(n+1),i} \end{pmatrix}. \tag{5.2}$$

The superscript $n+1$ denotes the time-step t_{n+1} and i the iterative procedure to resolve any fixed-point equation. The two fixed-point solvers follow the two steps illustrated in Figure 5.3. First, the computation of a Picard step (fixed-point iteration denoted by $\sim$), and second, an acceleration process to stabilize the fixed-point iteration. The acceleration can be a constant and adaptive Aitken under-relaxation or a quasi-Newton variant. The acceleration techniques' role in stabilizing and accelerating the FSI coupling is pointed out in [140].

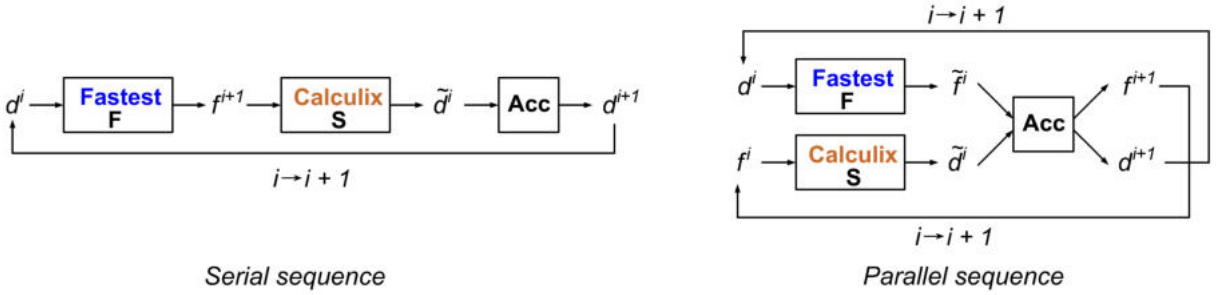


Figure 5.3.: Flow chart for the serial and parallel fixed-point solvers

Regardless of the selected sequence or acceleration method, the iteration loop is repeated until a **convergence criterion** is reached. The criterion can be an absolute or relative convergence for one or both couple variables. Depending on the choice, the coupling loop executes a different number of iterations. So, these settings directly influence the coupling performance and the overall stability of the solution. Here, the relative convergence criteria for both displacements and forces are usually used as is recommended in [140, 212]. These criteria compare the last iteration with the current Picard step before the acceleration process,

$$\text{Serial sequence} \quad \frac{\left\|\tilde{d}^i - d^i\right\|}{\left\|\tilde{d}_i\right\|} < \varepsilon_{rel}, \quad \text{and} \quad \frac{\left\|f^{i+1} - f^i\right\|}{\left\|f^{i+1}\right\|} < \varepsilon_{rel}\ , \tag{5.3}$$

$$\text{Parallel sequence} \quad \frac{\left\|\tilde{d}^i - d^i\right\|}{\left\|\tilde{d}_i\right\|} < \varepsilon_{rel}, \quad \text{and} \quad \frac{\left\|\tilde{f}^i - f^i\right\|}{\left\|\tilde{f}^i\right\|} < \varepsilon_{rel}\ . \tag{5.4}$$

In addition, the coupling variables can be first or second-order **extrapolated** at the beginning

of the time-step to get a better initial guess and accelerate the computation.

5.2.3. The acceleration techniques

Generalizing the fixed-point equations with an expression of the form

$$x = H\left(x\right), \tag{5.5}$$

the Picard step results in $\tilde{x}^i = H\left(x^i\right)$, and the acceleration by under-relaxation modifies the fixed-point iteration as

$$x^{i+1} = \omega^i \cdot \tilde{x}^i + \left(1 - \omega^i\right) \cdot x^i \quad \text{with} \quad 0 < \omega^i \leq 1. \tag{5.6}$$

The parameter ω^i can be a constant value defined by the user, or a dynamic Aitken [92] value given by

$$\omega^i = -\omega^{i-1} \frac{\left(R^{i-1}\right)^T \left(R^i - R^{i-1}\right)}{\|R^i - R^{i-1}\|_2^2} \quad \text{with the residual} \quad R^i := \tilde{x}^i - x^i. \tag{5.7}$$

The under-relaxation is well-suited for straightforward and stable problems, while for complicated cases, the acceleration with quasi-Newton methods is more robust [40]. The quasi-Newton methods accelerate the fixed-point iteration by a subsequent Newton step:

$$x^{i+1} = \tilde{x}^i - J_{\tilde{R},i}^{-1} \underbrace{\left(\tilde{x}^i - x^i\right)}_{R^i}, \tag{5.8}$$

where $\tilde{R} = I - H^{-1}$ resulting from the substitution of the inversion of the Picard step in the residual $R^i = \tilde{x}^i - H^{-1}\left(\tilde{x}^i\right) \rightarrow R^i = \tilde{R}\left(\tilde{x}^i\right)$. The not available inverse Jacobian $J_{\tilde{R},i}^{-1}$ is approximated based on the secant equation derived from the fixed point iteration,

$$\underbrace{\Delta\tilde{x}^i}_{W_i} = -J_{\tilde{R},i}^{-1} \underbrace{\tilde{R}\left(\tilde{x}^i\right)}_{V_i} \quad \text{with} \quad x^{i+1} = \tilde{x}^i + \Delta\tilde{x}^i, \tag{5.9}$$

and using the information from the past iterations stored in the matrices

$$\begin{aligned} W_i &= \left[\Delta\tilde{x}^i, \Delta\tilde{x}^{i-1}, \Delta\tilde{x}^{i-2}, \ldots, \Delta\tilde{x}^1\right], \quad \text{with} \quad \Delta\tilde{x}^k = \tilde{x}^k - \tilde{x}^{k-1}, \\ V_i &= \left[\Delta R^i, \Delta R^{i-1}, \Delta R^{i-2}, \ldots, \Delta R^1\right], \quad \text{with} \quad \Delta R^k = R^k - R^{k-1}. \end{aligned} \tag{5.10}$$

preCICE supports two methods to compute the Jacobian. The first is the Anderson acceleration [49], called **interface quasi-Newton** (IQN) [18, 222] approach, in which the Jacobian estimation is based on the minimal Frobenius norm and is written as

$$J_{\tilde{R},i}^{-1} = W_i \left(V_i^T V_i\right)^{-1} V_i^T. \tag{5.11}$$

The second is the generalized Broyden **multi-vector quasi-Newton (IMVJ)** [16, 126] approach based on a minimization of the distance between $J_{\tilde{R},i}^{-1}$ and approximate $J_{\tilde{R}}^{-1,(n)}$ from the last

time-step. The IMVJ estimate the Jacobian as

$$J_{\tilde{R},i}^{-1} = J_{\tilde{R}}^{-1,(n)} \left(W_i - J_{\tilde{R}}^{-1,(n)} V_i \right) \left(V_i^T V_i \right)^{-1} V_i^T. \tag{5.12}$$

In the IQN approach, old time-steps can be included in the matrices V_i and W_i to achieve fast convergence. However, too much old information can result in ill-conditioning, requiring filters to improve the stability and performance [15, 78].

5.2.4. Coupling configuration

The coupling library preCICE uses an *xml* file typically named **precice_config.xml** to configure the multiphysics simulation. In this file, one specifies first the Solver used in the coupled simulation, in this case, FASTEST and CalculiX, the coupling data that they exchange, Forces and Displacements, and the coupling meshes. The meshes Fluid_Nodes and Fluid_Centers for FASTEST, and Structure_Nodes for CalculiX. Likewise, one configures the numerical methods used for the data mapping in the write and read direction. Finally, one sets the fixed-point solver, including the specifications of the time-step size, the maximum time, the data exchange direction, the convergence measures, the maximum number of iterations per time window, and the acceleration technique. Appendix D is an example of this configuration file, and the reader can find an overview of the complete configuration in `https://www.precice.org/configuration-overview.html`.

5.3. FASTEST Adapter

The FASTEST Adapter is structured in a separate module written in Fortran language to avoid direct modifications to the Solver code and call all preCICE APIs from this module only, thus facilitating code maintenance. The adapter module consists of three public subroutines *initprecice, unsteadyfmgprecice*, and *finalizeprecice*, and two internal process *readprecice* and *writeprecice*. The **initprecice** subroutine establishes communication channels and data structures of preCICE, defining coupling meshes and getting mesh and data identifiers. It is called in the code initialization part. The second subroutine, **unsteadyfmgprecice**, calls the preCICE APIs to control the time loop and the implicit iterative FSI loop and calls **readprecice** and **writeprecice** to read displacements and to send forces, respectively. Finally, the **finalizeprecice** subroutine closes the communication channels and stops the FSI coupling when the simulation's maximum time is achieved. The FASTEST Adapter's flowchart is schematized in Figure 5.4, where the green words indicate the new processes added for computing FSI problems and the green frames where the preCICE APIs are used. The description of the sequence is below.

After establishing communication and setting the coupling data, at each time-step, the program executes the following steps:

1. Call *ongoing* API to inform FASTEST of the start of a new time-step or the end of the coupling.
2. Save the current state of the variables: velocity, pressure, volume fraction, and grid coordinates as the **old state**.
3. Set the last time solution of the fluid field as the initial guess solution for the new time-step.

4. Computes the local Courant number.

5. If the adaptative time-step is active, adjust the time-step Δt_{solver}..

6. Increment the time with the minimum time-step size between Δt_{solver}. and $\Delta t_{coupling}$.

7. Call the subroutine *readprecice* to read the displacements **d** from the coupling *Fluid_Nodes* mesh.

8. Deform the mesh employing the algebraic and elliptic approaches detailed in [180, 186]. The grid deformation follows three steps. First, the block edges distort. Second, the faces distort, and third, the current condition of the faces is the boundary conditions to distort the internal grid.

9. Compute the wall velocity field at the FSI interface and set it as a Dirichlet boundary condition. The velocity is calculated as the variation of the boundary CV-centers on time using an implicit Euler or BDF2 time scheme.

10. Solve the VoF transport equation iteratively (loop-a in the flowchart of Figure 4.4) to find the volume fraction field α for the new grid configuration.

11. Update the nodal and face density and viscosity properties and compute the surface tension.

12. Solve the discrete form of the space conservation law (Section5.4) to compute the additional convective fluxes introduced by the ALE formulation.

13. Compute the enhanced SIMPLE algorithm by PRIME steps to find the new velocity and pressure fields.

    The flow variables $(\mathbf{v}, p)$ converge? **yes**: go to **14**, **no**: return to **13**

14. Update the relative volumetric flux $F^{\alpha}_{f,g}$ used to compute the volume fraction field.

15. Determine the Surface forces **F**.

16. Call *writeprecice* to write the forces in the *Fluid-center* mesh and send to preCICE.

17. Send to preCICE the time-step size used in FASTEST, Δt_{solver}.

18. Read from preCICE, the time-windows coupling size $\Delta t_{coupling}$, and the *ongoing* and *checkpoint* variables' status. The first variable, *ongoing*, controls the advance in time, and the second returns the FSI convergence status.

    FSI coupling variables $(\mathbf{d}, \mathbf{F})$ converge? **yes**: return to **1** to advance in time, **no**: go to **19**.

19. Reset the variables to the old state and return to **7**.

In problems without free surface flows, the algorithm does not execute the two-fluid flows calculations given by steps **5**, **10**, **11**,**14**.

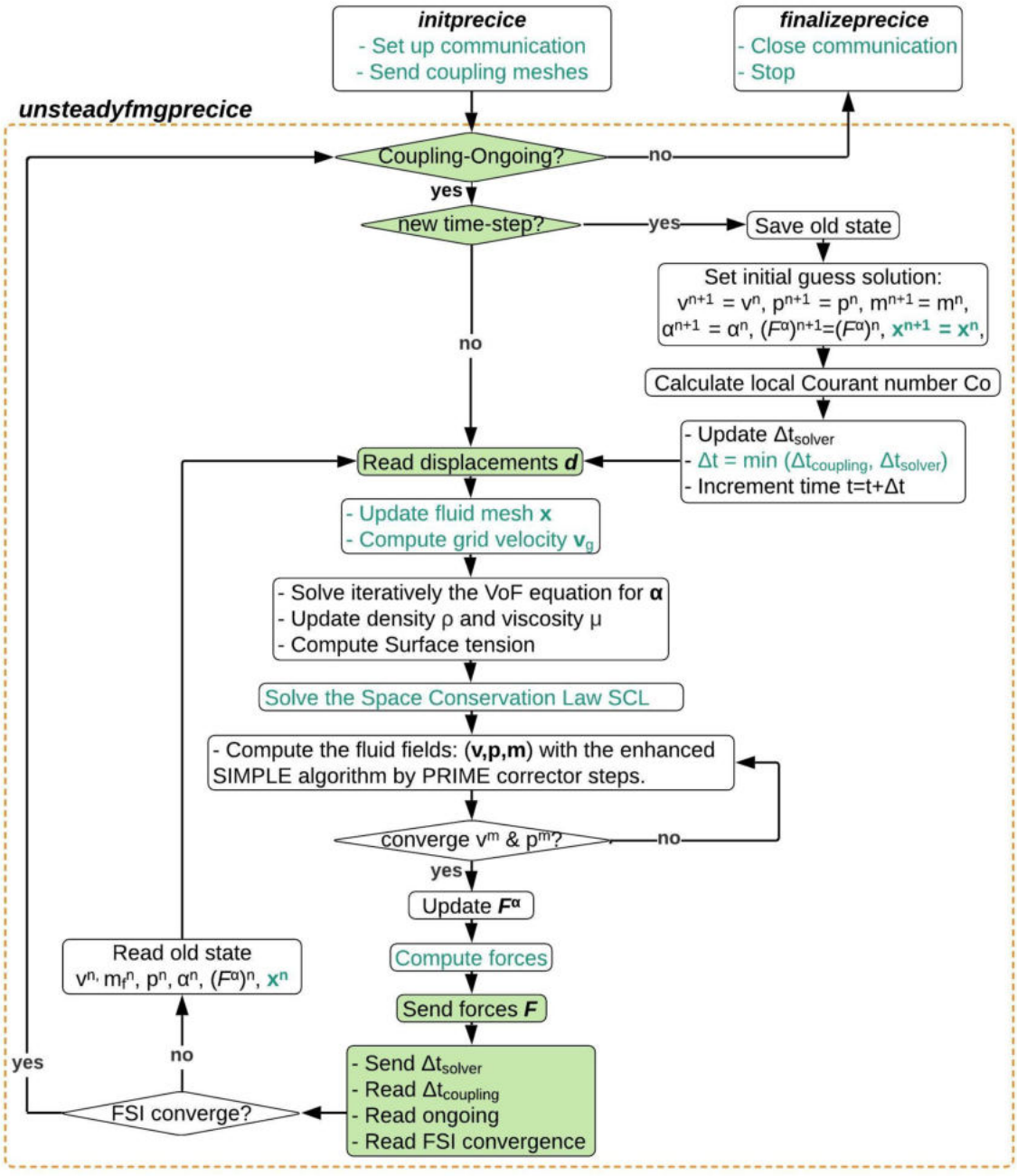


Figure 5.4.: Flowchart of FASTEST Adapter

5.4. The discrete form of the space conservation law

Since the flow solver is developed in the context of the ALE framework to address the moving domains, the fulfillment of the space conservation law (SCL), Eq. (2.10), is mandatory to force mass conservation. Here, the discrete form of the SCL is applied. It is developed in reference [41] in the context of FVM and has the form

$$\sum_{f(P)} \frac{\partial \left(\delta V_f\right)}{\partial t} = \sum_{f(P)} \left(\mathbf{v}_\mathrm{g} \cdot \mathbf{n}\right)_f \delta S_f. \tag{5.13}$$

This equation states that the change in volume of each CV in time must be equal to the volume swept by the CV-faces during the time interval. Solving the temporal derivative of Eq. (5.13) and multiplying it by the density on the face, we find, on the right-hand side, the face grid flux $\dot{m}_{g,f}$ at $t = t_{n+1}$, which is the additional convective flux introduced by the ALE formulation.

Applying the forward Euler time scheme, the face grid flux results in

$$\dot{m}_{g,f}^{n+1} = \rho_f^{n+1} \frac{\Delta \delta V_f^n}{\Delta t_n}, \tag{5.14}$$

or using the BDF2 time scheme with variable time-step, it is

$$\dot{m}_{g,f}^{n+1} = \rho_f^{n+1} \left[k_P^{n+1} \left(\Delta \delta V_f^n\right) - k_P^{n-1} \left(\Delta \delta V_f^{n-1}\right) \right], \tag{5.15}$$

where $\left(\Delta \delta V_f\right)^n = \left(\delta V_f^{n+1} - \delta V_f^n\right)$, and $\left(\Delta \delta V_f\right)^{n-1} = \left(\delta V_f^n - \delta V_f^{n-1}\right)$ represent the volume swept by the cell face f during the time Δt_n and Δt_{n-1}, respectively. The k coefficients are the same as already defined in the transient discretization of the momentum equation.

Likewise, the face grid volumetric fluxes $F_{f,g}^{\alpha}$ used in the VoF transport equation are computed. Only for these quantities, the density is omitted.

5.5. Validation of the fluid-structure interaction coupling

The validation of the redesigned fluid-structure interaction coupling algorithm is performed, solving the FSI3 benchmark proposed by Turek and Hron [207]. This test case was successfully simulated with various numerical approaches in [206], and nowadays, this is a well-established standard validation test in the FSI community, e.g., [19, 180, 212]. The FSI3 benchmark involves an incompressible Newtonian and laminar flow interacting with an elastic bar attached to a fixed cylinder, resulting in self-induced oscillations of the structure. Figure 5.5 shows the geometrical configuration and boundary conditions of the problem. A parabolic inflow velocity profile is prescribed at the left boundary, while at the right boundary, an outflow with zero gradient condition is assumed. The top and bottom of the channel and the complete structure's surface are set as no-slip boundaries. For details about the interaction problem's overall setup, see [207].

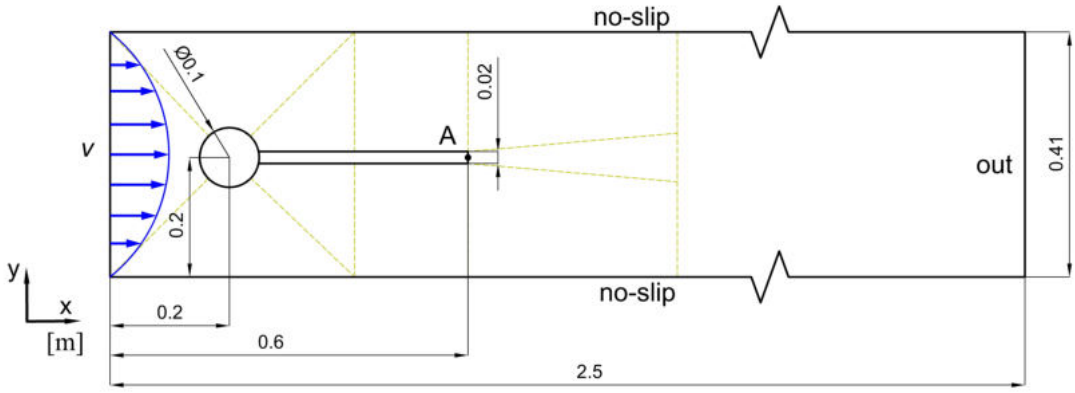


Figure 5.5.: The geometric configuration and boundary conditions of the FSI3 benchmark test case. A is the control point. The dashed lines indicate the distribution of the blocks in the fluid grid

The validation is carried out by solving three grid levels with a refinement factor of 2 in the x and y direction for the fluid and structural subdomains. Once the dynamic movement of the bar converges in periodic oscillations, the mean values, amplitudes, and frequencies of the displacement of point A and the total draglift forces around the cylinder and elastic bar are compared with the reference values presented in [206].

The fluid domain is discretized by a block-structured grid that consists of 11 blocks of hexahedral control volumes distributed as is shown with dashed lines in Figure 5.5. The coarse mesh consists of 15232, the medium mesh of 60928, and the fine mesh of 243712 control volumes. As this case is 2D and the fluid solver, FASTEST, is a 3D solver, a quasi-two-dimensional setup is achieved by setting slip boundary conditions in the z-direction and using only a control volume of 0.01 m thickness. Furthermore, the velocity and distortion in the z-direction are set to zero. The time integration of the equations is performed with the BDF2 scheme. The internal convergence in the fluid solver, FASTEST, is set to 10^{-7}, and the SIMPLE mode of the pressure-velocity coupling loop is limited to a maximum of 10 iterations. The grid remeshing is handled with the linear transfinite interpolation (TFI) method. Information about grid movement techniques and their influence in fluid-structure interaction computations can be found, for example, in [241].

The structural part is discretized by a structure grid of fully integrated linear eight-node brick elements (C3D8 element type in CalculiX). The shape function is described in [112]. The coarse, medium, and fine mesh elements are 32×2, 64×4, 128×8, respectively. All nodes are limited to two degrees of freedom (translation in x and y). The bar is modeled as Saint-Venant Kirchhoff material model considering the nonlinear geometric effects. The dynamic analysis is conducted with the direct integration analysis that integrates the equation of motion in time with the classical Newmark method. The convergence criteria used are the default values in CalculiX.

For the FSI coupling, an implicit serial coupling is set with a constant under-relaxation of 0.2, a maximum number of FSI iterations per time-step of 50, and the relative convergence threshold of 10^{-6} for displacements and 10^{-2} for forces. After testing, these values were chosen because a stricter convergence limit for the forces slows down the simulation too much without significantly improving the results. The three grid levels are run with a constant time-step 2×10^{-3} s for the two

solvers. The grids for the fluid and structure part are no matching grids at the coupling interface, so the displacement and forces at the interface are mapped with the radial basis function method without considering the z-direction. Also, a second-order extrapolation for the displacements is applied to accelerate the calculation. The *precice_config.xml* file with the complete coupling settings for this case is given in Appendix D.

Initially, the flow is developed without the structural model until this converges to a periodic regime (16 seconds), then this solution is imposed as an initial condition for the FSI case. The simulation is run for 4 seconds, where the first 2 seconds show the transient phase's influence, and the last part shows a periodic oscillations behavior. Figure 5.6 shows the velocity contour plots in the x-direction, the coarse fluid mesh distortion, and the structure's position at times $t = 19.7$ s and $t = 19.8$ s.

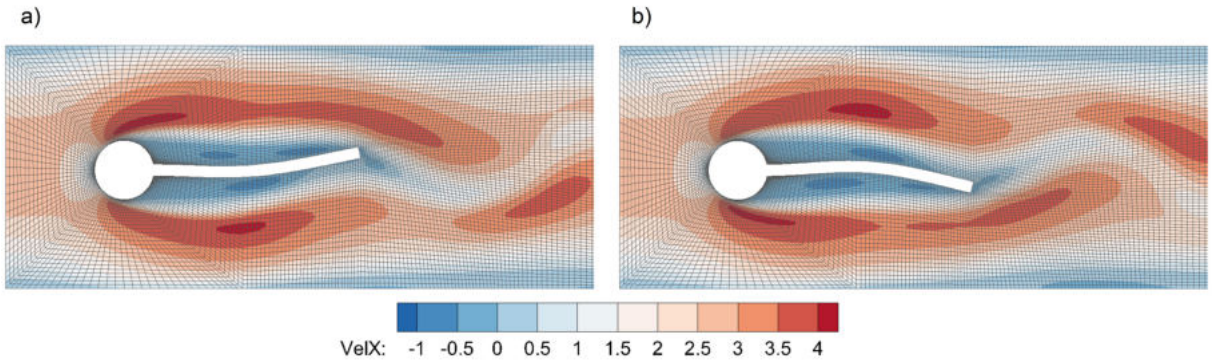


Figure 5.6.: x-velocity profile and coarse grid distortion for the FSI3 benchmark case for a) $t = 19.7$ s and b) $t = 19.8$ s

Figure 5.7 displays the displacements in a) x and b) y-direction, as well as the total c) lift and d) drag forces for the three grid levels over time from 19.5 s to 20 s, while Table 5.1 lists the relevant benchmark values computed from the second oscillation after $t = 19.5$ s. The reference values are the results presented by Turek/Hron in [206]. The data for the curves was downloaded from `http://www.featflow.de/en/benchmarks/cfdbenchmarking/fsi_benchmark.html`.

The coarse grid results are more discrepant to the reference value due to a too coarse spatial discretization. However, the medium and fine grid levels are close to the reference results. In general, the maximum differences between the finest grid level results and the reference values are around 4%.

Finally, other important points of a partitioned FSI coupling algorithm are efficiency, robustness, and stability. They are closely linked to coupling interfaces meshing, conforming or non-conforming meshes, thus implicitly the mapping strategy and the method used to solve the fixed-point iterations process by the coupling tool preCICE [19]. Here, the results obtained with a constant under-relaxation of 0.2 were presented. It required around 29 FSI iterations per time-step for the coarse grid and 36 for the medium and fine grid. However, other settings can be used in the preCICE configuration that either improve or reduce the efficiency and stability of the coupling. For instance, one of the configurations tested involved an under-relaxation factor of 0.05, a maximum number of FSI iterations of 200, and a convergence criterium for the displacement of 10^{-4}. It required around 75 FSI iterations per time-step for the coarse and medium grid, and

the fine grid becomes unstable. A smaller time-step of 0.001s was used for a stable simulation that required about 126 FSI iterations per time-step. Thus, this alternative coupling configuration doubled the computational time for the coarse and medium and quadrupled it for the finest grid level.

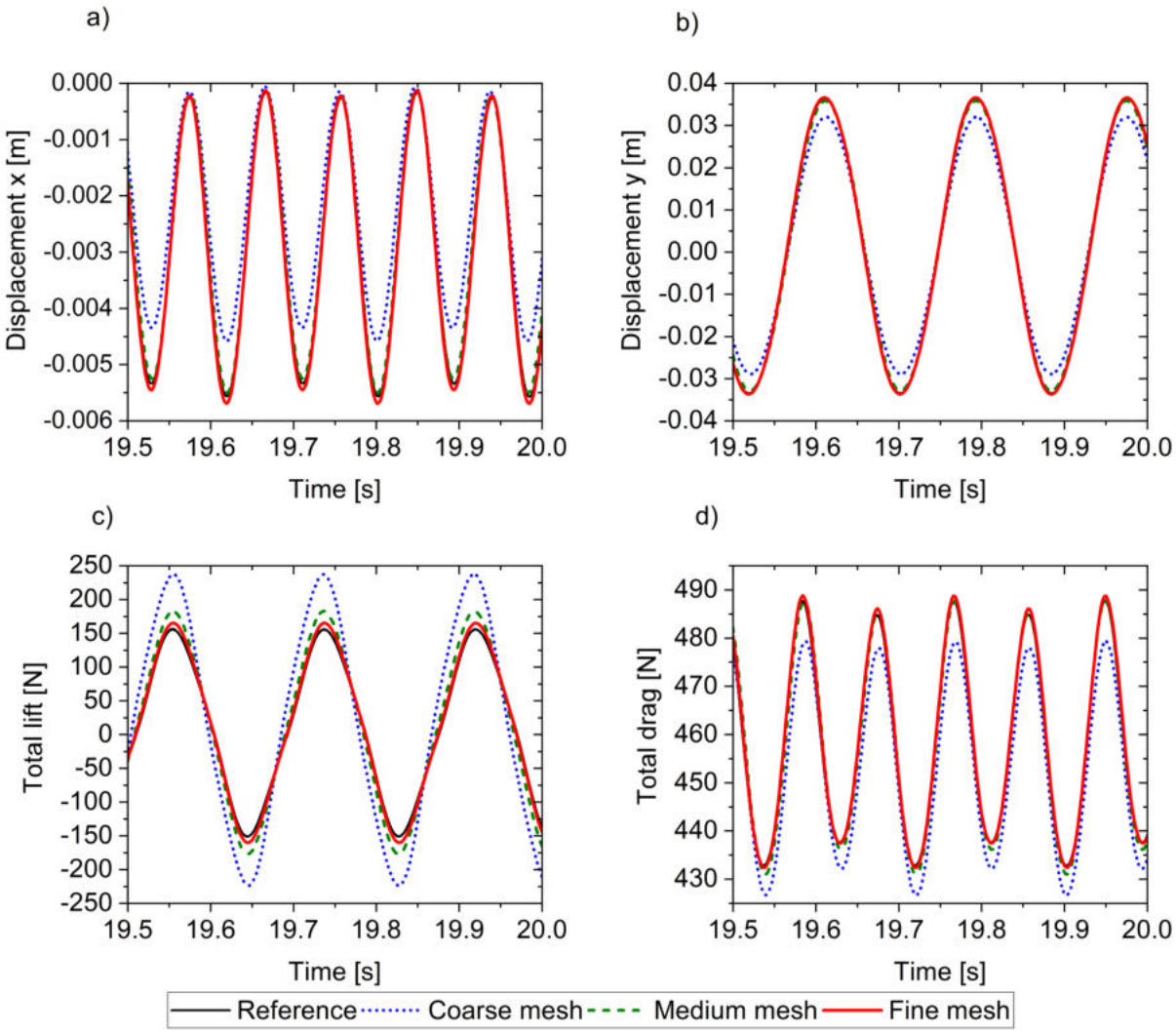


Figure 5.7.: Displacement and force values for the FSI3 benchmark test case for the three grids levels and the reference values [206]. a) x and b) y-displacement of point A, total c) lift and d) drag forces on the cylinder with elastic bar

Conclusions:

The validation test case, the FSI3 benchmark, was successfully simulated using the newly implemented FSI coupling algorithm. The displacement and forces agree with the reference results. Consequently, the algorithm works properly and can be used for other FSI test cases. Concerning

Grid level	ux(A) $[10^{-3}]$	uy(A) $[10^{-3}]$	LIFT [N]	DRAG [N]
Coarse	$-2.32 \pm 2.25\,[10.75]$	$1.57 \pm 30.5\,[5.68]$	$6.61 \pm 230.43\,[5.56]$	$453.01 \pm 26.31\,[10.87]$
Medium	$-2.81 \pm 2.69\,[10.64]$	$1.49 \pm 34.3\,[5.49]$	$2.75 \pm 179.61\,[5.49]$	$459.32 \pm 28.33\,[11.11]$
Fine	$-2.91 \pm 2.78\,[10.87]$	$1.48 \pm 35.1\,[5.46]$	$2.60 \pm 162.59\,[5.43]$	$460.58 \pm 28.19\,[10.64]$
Turek/Hron	$-2.88 \pm 2.72\,[10.93]$	$1.47 \pm 35.0\,[5.46]$	$2.50 \pm 153.9\,[15.46]$	$460.50 \pm 27.47\,[10.93]$

Table 5.1.: Displacement values at point A and lift-drag forces on the cylinder with the elastic bar for the FS3 benchmark test case. The values are given as the mean ± amplitude [frequency] and are computed from the second oscillation after $t = 19.5$ s.

efficiency and stability, several parameters can be tested in the coupling tool. Nevertheless, it was not the objective of this validation analysis. The reader can find other ideas to solve this test case, e.g., in [68] or [140], where matching grids were used at the FSI interface and the quasi-Newton methods to accelerate the fixed-point iteration reducing the computational time considerably.

5.6. FSI involving free surface flows applications

5.6.1. Dam-break with elastic obstruction

The collapsing column of water hitting an elastic structure problem is a modification of the experimental case of Koshizuka [108] presented in Section 4.10.4, by replacing the rigid obstacle with an elastic structure. This case was firstly solved by Walhorn et al. [226] using a monolithic FSI approach. Although there are still no experimental results of this example, nowadays, it is a typical validation test for FSI algorithms that involve free surface flows (see, e.g., [15, 91, 101, 121, 165, 168, 249]. The problem setup and the initial condition in one of the computational domains are shown in Figure 5.8. It consists of a 292 mm high column of water that collapses under gravity, hitting an 80×12 mm^2 elastic bar. The tank is open at the top, and slip boundary conditions are imposed at the walls. The surface tension effects are ignored due to the considerable length scales of the problem.

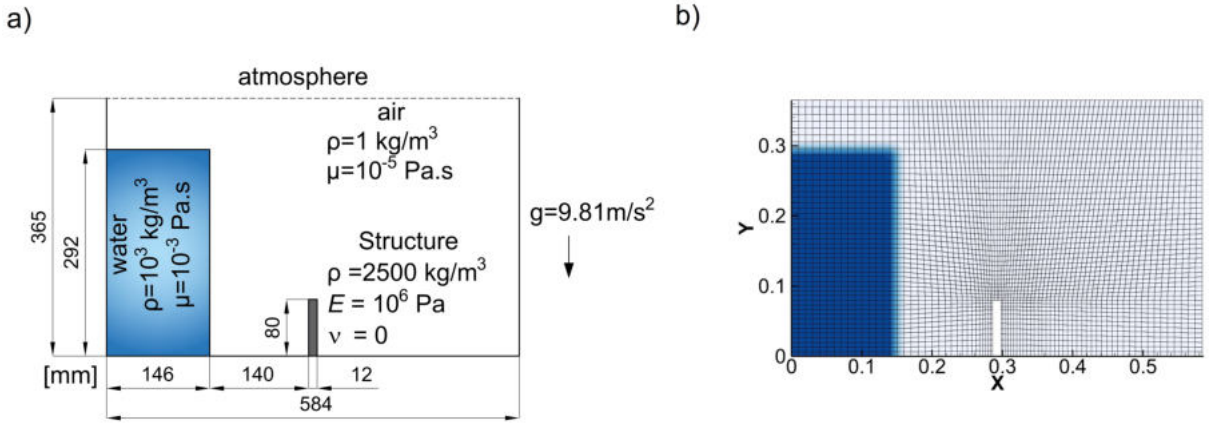


Figure 5.8.: Dam-break with elastic structure a) geometric configuration, b) computational fluid domain (grid 3114 CVs)

Two grid refinement levels are used to simulate this case. The fluid domain is discretized with a coarse grid consisting of 3792 CVs and a fine grid of 15168 CVs (see mesh distribution in Figure 5.8b). While the corresponding structure domain is formed by 24 and 96 incompatible mode eight-node brick elements (C3D8I). The time-step for the coarse level is 10^{-3} s and for the fine level 5×10^{-4} s. The relative convergence tolerance for displacements and forces at the coupling interface is set to 10^{-4} and 10^{-2}, respectively. In the fluid solver, the enhanced SIMPLE algorithm with two PRIME steps solves velocity and pressure fields. The CUIBS scheme numerically implemented with the MNWF method captures the free surface flow dynamics. The temporal discretization of the fluid domain is carried out with the BDF2.

The structure solver is set to model the elastic structure with a Saint Venant-Kirchhoff material that considers the nonlinear geometric effects in the strain tensor. A direct integration procedure of the equations of motion in time with the Newmark method and constant time increment computes the structure's response to dynamic fluid loads. The FSI coupling is solved with a serial implicit scheme accelerated for the coarse grids with a constant relaxation of 0.5 and the fine grids with Aitken under-relaxation with an initial value of 0.25. Figure 5.9 presents the snapshots at various instances of time for the fine grid level.

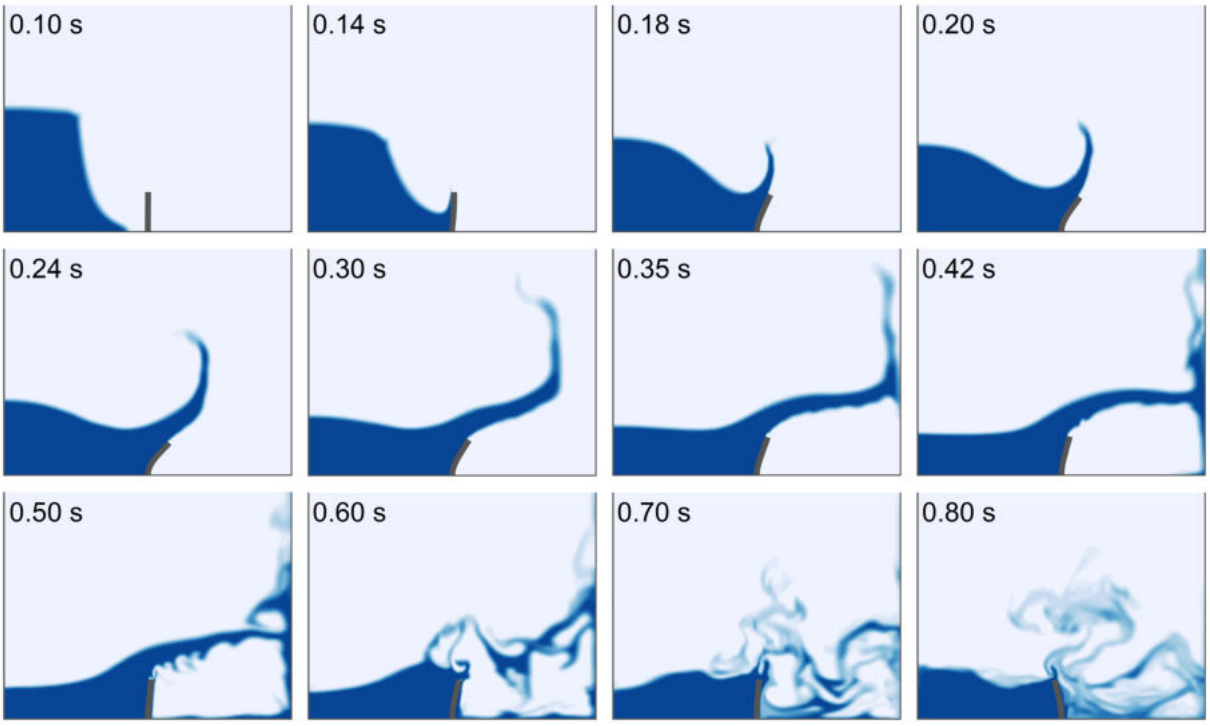


Figure 5.9.: Wave iteration with elastic structure at various times, grid 15168 CVs – 96 elements

The time history of the displacement of the elastic beam at the upper left corner during 0.85 s is depicted and compared with the available literature references in Figure 5.10. The water column has the same behavior as the previous calculation in Section 4.10.4 until the water wave reaches the elastic beam, then the wave impulse deforms it. The left upper corner of the beam first slightly deflects to the left when the wavefront impacts the lower part of the beam ($t \approx 0.14$ s) and moves

to the right while the wave rises. The maximum deflection occurs at $t \approx 0.24$ s and is about 4.5 cm. The wave continues to move to the right part of the domain, and the beam starts to gradually return to the original position. When the water impacts the wall ($t \approx 0.34$ s), the beam begins to oscillate, and the turbulent water at both sides of the beam damps the structure oscillations.

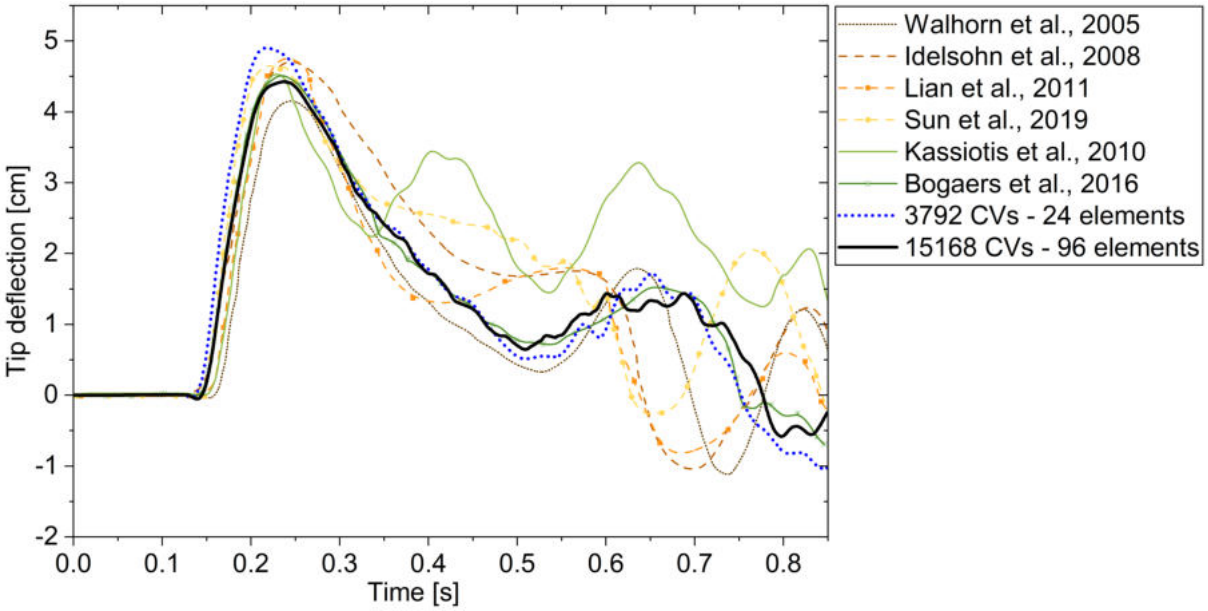


Figure 5.10.: History of the x-displacement of the upper left corner of the elastic structure for the dam break benchmark problem

There are no relevant differences between the two grid levels used to simulate this case. However, the results obtained by the proposed method coincide with the reference results before $t \approx 0.34$ s. This is when the water waves impact the right wall. Afterward, the discrepancy among the different numerical solutions increases because of the strong nonlinearities of high-velocity flows. The damping period depends on how violent fluid-structure interaction is solved. For instance, the calculations based on Lagrangian particle methods, Idelsohn et al. [91], Lian et al. [121], and Sun et al. [196], are more close to each other. The structure damps two times but with a different frequency from the initial results of Walhorn et al. [226].

In contrast, the present results are in perfect agreement with those published by Bogaers et al. [15]. The first and second deflections are very close to the curves computed by Walhorn et al. with the monolithic approach. The partitioned approach used by Bogaers et al. is based on the FVM, VoF method, and FEM as the proposed approach. Although Kassiotis et al. [101] also use the combination of the three methods, their results are different from the others.

According to the literature review, the structure's damping discrepancies are due to the tricky turbulent part of the case. However, I observed that the reference works used different structural material models to obtain the results, and maybe this is also a reason for disagreements. In order to find the influence of the material model, the coarse grid level is again simulated with three different structural material models: linear elastic, Saint Venant-Kirchhoff, and Neo Hooke. Figure 5.11 compares the results. When the liner material is set, the second deflection begins as the

water wave impacts the right wall. This curve is similar to the ones presented by Kassiotis et al.. In contrast, when using the Saint Venant-Kirchhoff material, the beam undergoes less damping, as shown in the present results and the ones of Bogaers et al. Using the Neo Hooke model, the damping begins earlier than with the Saint Venant-Kirchhoff material. The results are closer to the solutions obtained with the Lagrangian particle approaches, which assumed a Neo Hooke material model.

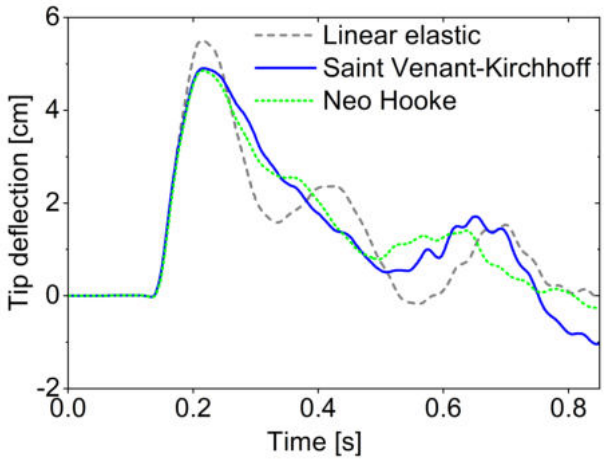


Figure 5.11.: History of the x-displacement of the upper left corner of the elastic structure for different elastic material models

Conclusions:

- The new approach for simulating free surface flows in interaction with flexible structures can correctly solve complex problems. The free surface profile and the deformation of the elastic beam are in accordance with the governing physics of the problem.
- Although the results presented here and the references agree qualitatively, there are some differences between those obtained by different methods. The results mainly depend on the method's capability to capture the violent free surface flows and the use of turbulence fluid models, which are not covered in this work. Besides, we observed that the chosen material model also influences the global solution. Hence, experimental results of this case would be necessary for complete validation of the numerical results.

5.6.2. Dam-break with an elastic plate

The dam-break with an elastic plate is a series of quasi-two-dimensional experiments accompanied by simulations carried out by Liao et al. [123] to study, similarly to the previous case, the dam break phenomenon in interaction with an elastic structure. Figure 5.12 illustrates the 2D configuration of the case. In the experiments, the water column of height H_0 is retained in the tank's left part by a gate. Once the gate opens, the water column collapses under gravity and flows to the tank's right side, impacting on a thin rubber plate, which begins to bend. The height H_0 takes different values that lead to different impact forces, which cause different initial deflections

of the elastic plate. Three LEDs (Marker 1-3) installed on the plate at different heights are used to track the structural deformation, and a high-speed digital video camera captures the free surface profile. The plate's severe deformation and the strong nonlinear free surface flow dynamics have turned this case into an excellent benchmark to evaluate the robustness of numerical FSI approaches (compare to [137, 193]).

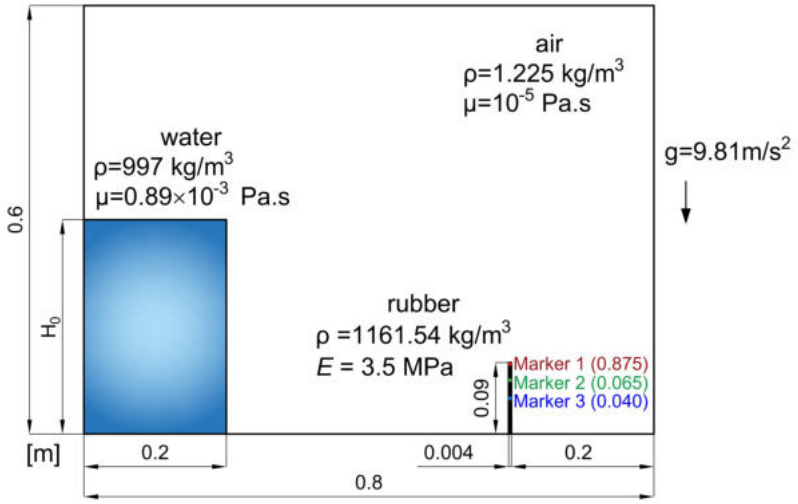


Figure 5.12.: Geometrical configuration and set up of the dam break with an elastic plate case

Here, two values of the initial water height, $H_0 = 0.2$ m and 0.3 m, are computed and contrasted with the benchmark results to validate the present numerical FSI framework. The case is addressed as a 2D scenario. Thus the plate is treated as a cantilever beam. The tank walls are considered no-slip boundary conditions, while the tank's top boundary remains open to the atmosphere. The displacement of the beam bottom is fixed to zero. The fluid domain is discretized with 23328 hexahedral CVs and the structural domain with 26 brick C3D8I elements. The simulation is carried out up to 1.0 s with a fixed time-step of 2×10^{-4} s.

The two-fluid flow dynamics is solved with the pressure-velocity coupling algorithm set in SIMPLE mode, and the CUIBS interface capturing scheme tracks the free surface interface. The rubber beam of density 1161.54 kg/m^3, Young's modulus 3.5 MPa, and the Poisson's ratio 0.475 is modeled with the Mooney-Rivlin hyperelastic material model ($C_{10} = 5.98 \times 10^5$ Pa, $C_{01} = 0$ Pa, and $D_1 = 8.57 \times 10^{-8}$). The FSI coupling is accelerated with a constant under-relaxation of 0.3.

Figures 5.13 and 5.14 show the evolution of the displacement in the x direction of the tip of the elastic beam for two different water column heights. The present numerical results are in good agreement with those obtained by Liao et al. [123]. The numerical method captures the main features of the phenomenon observed in the experiment, even the negative displacements due to the initial wave impact. One can observe that the numerical beam in the case of $H_0 = 0.2$ m underestimates the two maximum displacements reported in the experiment, especially for Marker 1. As pointed out by the authors in [123], for the height $H = 0.2$ m, after the impact of water on the plate, a trapped air bubble is formed in the experiment. It plays a vital role in the subsequent flow evolution, including the hydrodynamic loading on the elastic structure and its

subsequent elastic response. We can observe these differences between the experimental and numerical results in the snapshots comparison presented in Figure 5.15. Furthermore, in this work, the complex turbulent flow structures formed in this case are not treated, and maybe this is also a reason for the slight differences with the reference results.

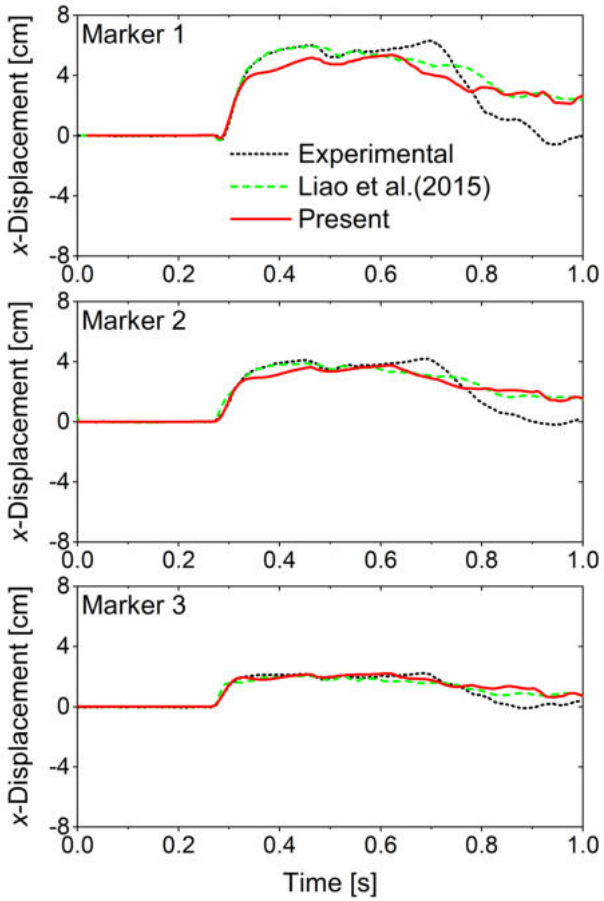


Figure 5.13.: Comparisons of displacement in x-direction of the dam break with an elastic plate case, when $H_0 = 0.2$ m

In contrast, when $H_0 = 0.3$ m, the maximum displacement is similar to the experiment, and the computed x-displacements coincide in most of the numerical references. Although for the last 0.2 seconds, the present results are nearer to the experimental results than those presented by Liao et al. Their prediction of the last position of the beam is 50% higher than the experiment. Also, a significant amount of the higher modes of vibration of the elastic beam presented in the experiment are well captured by the present approach. The agreement of the present results with the experiments for this height can also be observed in Figure 5.16, which shows a qualitative comparison between them at different time instants.

Finally, regarding computational time, there are no significant differences between the two cases. Both require almost eight hours using eight processors for the fluid part and one for the structural part. Around five FSI iterations per time-step are necessary to achieve relative

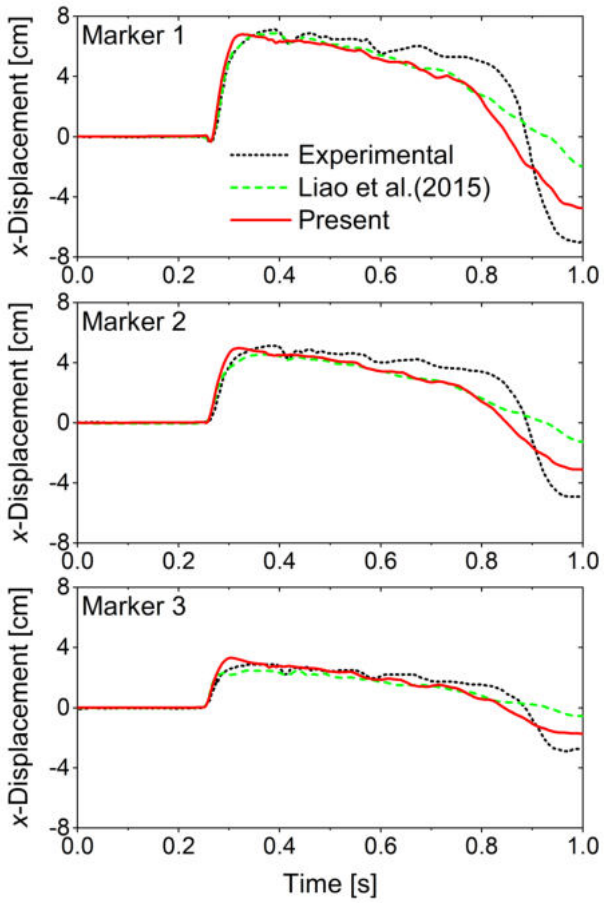


Figure 5.14.: Comparisons of displacement in x-direction of the dam break with an elastic plate case, when $H_0 = 0.3$ m

convergence criteria of 10^{-4} for displacements and forces at the FSI interface. It is noteworthy that even though the maximum Co numbers for these simulations were 1.30 ($H = 0.2$ m) and 2.27 ($H = 0.3$ m), the calculations remain stable, and the free surface interface has not exhibited excessive diffusion. Thus, the approach is a suitable combination of accuracy and efficiency.

Conclusions:

As shown in these two scenarios, the proposed FSI approach involving free surface flows can reproduce the phenomenon with good accuracy in a reasonable time. From the computational point of view, the FSI approach could solve the problems without numerical instabilities, even for high nonlinearities. However, the violent turbulence flow reported in the experiments cannot be predicted precisely by the present numerical approach. The improvement of the results requires the inclusion of turbulence models outside of this thesis scope.

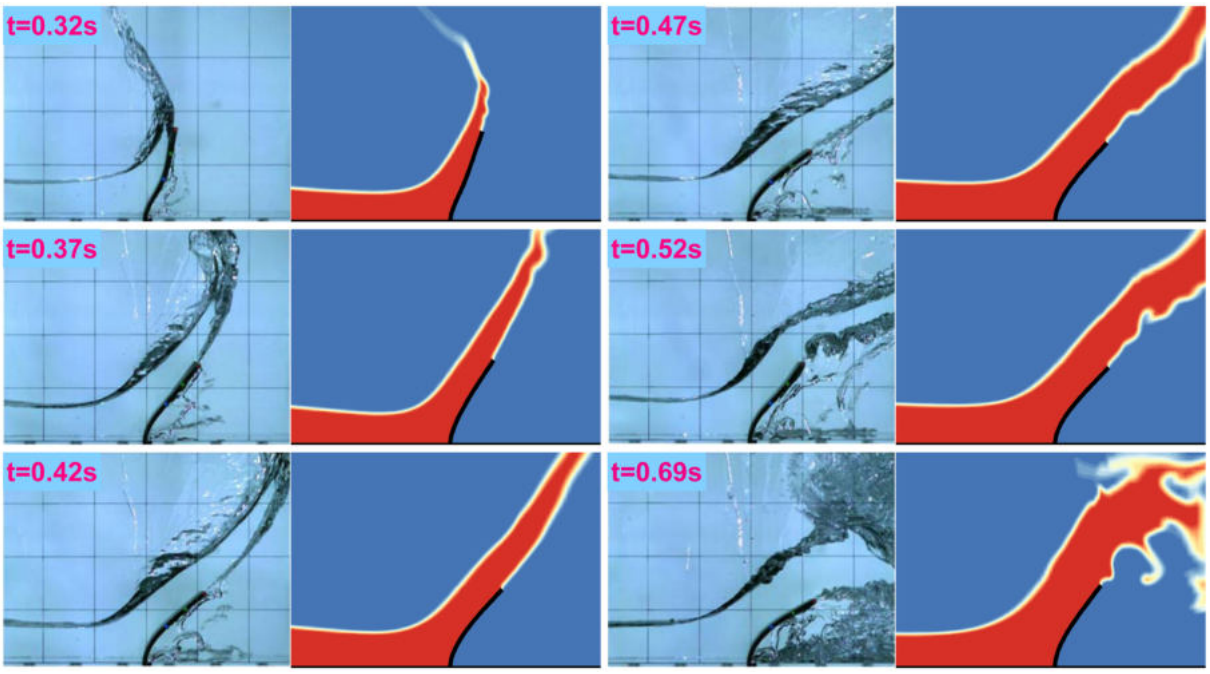


Figure 5.15.: Free surface profile and structural deformation at different times (Experiment vs. Numerical results, $H_0 = 0.2$ m

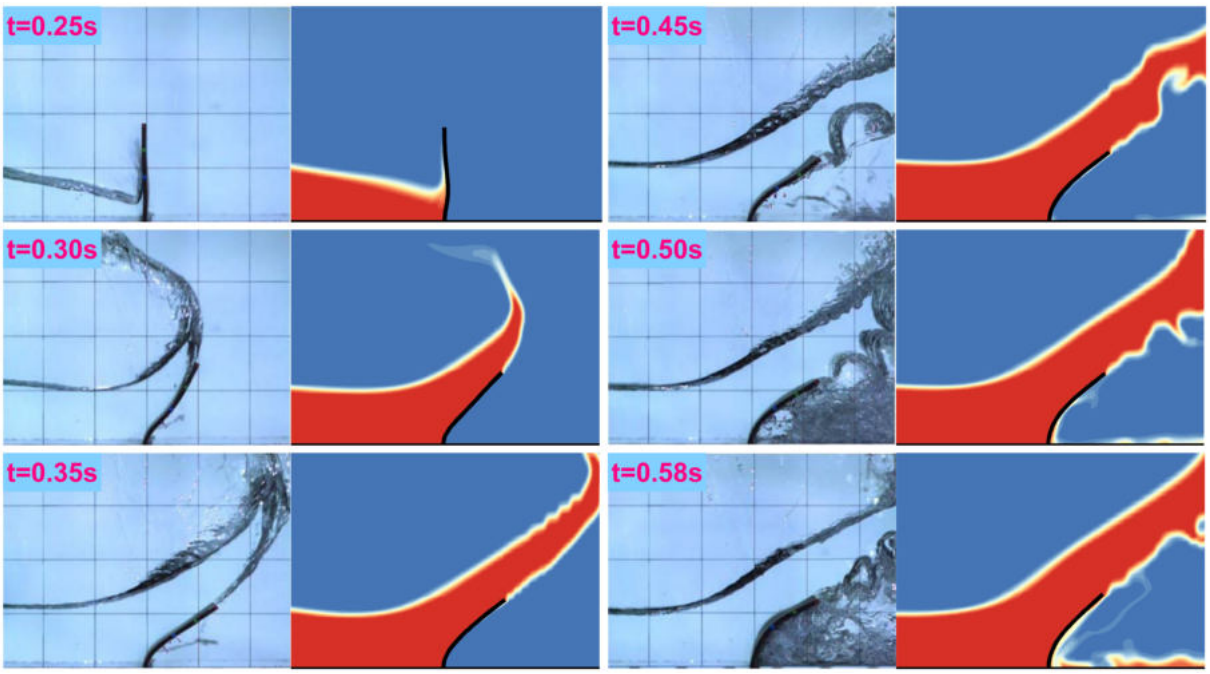


Figure 5.16.: Free surface profile and structural deformation at different times (Experiment vs. Numerical results, $H_0 = 0.3$ m

5.7. Dam-break with a moored deformable platform

The last application aims to show the versatility of the present approach for simulating more practical problems. The stability of a moored deformable floating structure under the impact of a solitary wave is discussed. The case was introduced in reference [194] as an example test of deformable floating structures. Figure 5.17 illustrates the geometry and parameters of the test case. A tank open to the atmosphere contains a water dam that collapses under gravity and forms a solitary wave that impacts and deforms a platform moored on the tank's bottom by two cables. The tank in [194] has a width of 56 cm and includes a damping district on the tank's right a damping district to avoid the reflected wave. Here, the damping district is modeled by elongating the width to 100 cm. The cables' spring constant is 400 N/m, and they are assumed to be very thin and light, so their interaction with the fluids is ignored. The platform is analyzed for two scenarios with different Young's modulus, 0.4 MPa, and 40 MPa, to observe the influence of the flexibility in the structure response.

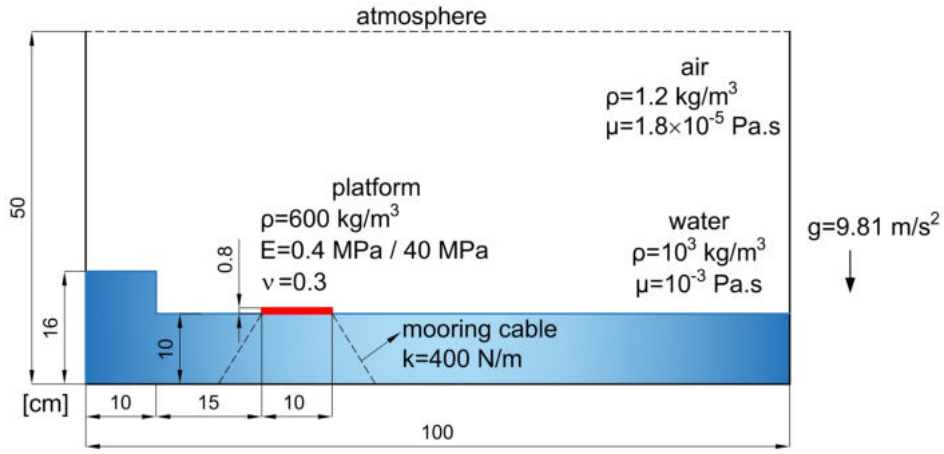


Figure 5.17.: Geometric configuration of the dam-break with a moored deformable platform test case

The computational two-fluid flow domain consists of 10256 CVs biexponentially distributed and nested near the structure. The tank walls are defined as slip boundary conditions. The platform is model as a cantilever beam formed by ten three-node beam elements (element type B32R in CalculiX), and the deformation follows the Saint Venant-Kirchhoff material model. Two spring elements set in the structural solver define the two cables. They are attached to the first and the last node of the beam and fixed to the tank's bottom. The serial coupling scheme accelerated by dynamic Aitken under-relaxation with an initial value of 0.25 solves the FSI calculation. The relative convergence tolerance for the displacements is 10^{-6}, and for the forces, 10^{-4}. The computations are run up 1 s with a constant time-step of 0.001 s.

Figure 5.18 shows the free surface profiles interacting with the flexible platform ($E = 0.4$ MPa) and the rigid one ($E = 40$ MPa) in time series. We can observe that the general motion of the structure is very similar for the two cases. However, the more flexible platform displays more bending deformation than the rigid one during the wave impact (0.20 s to 0.45 s).

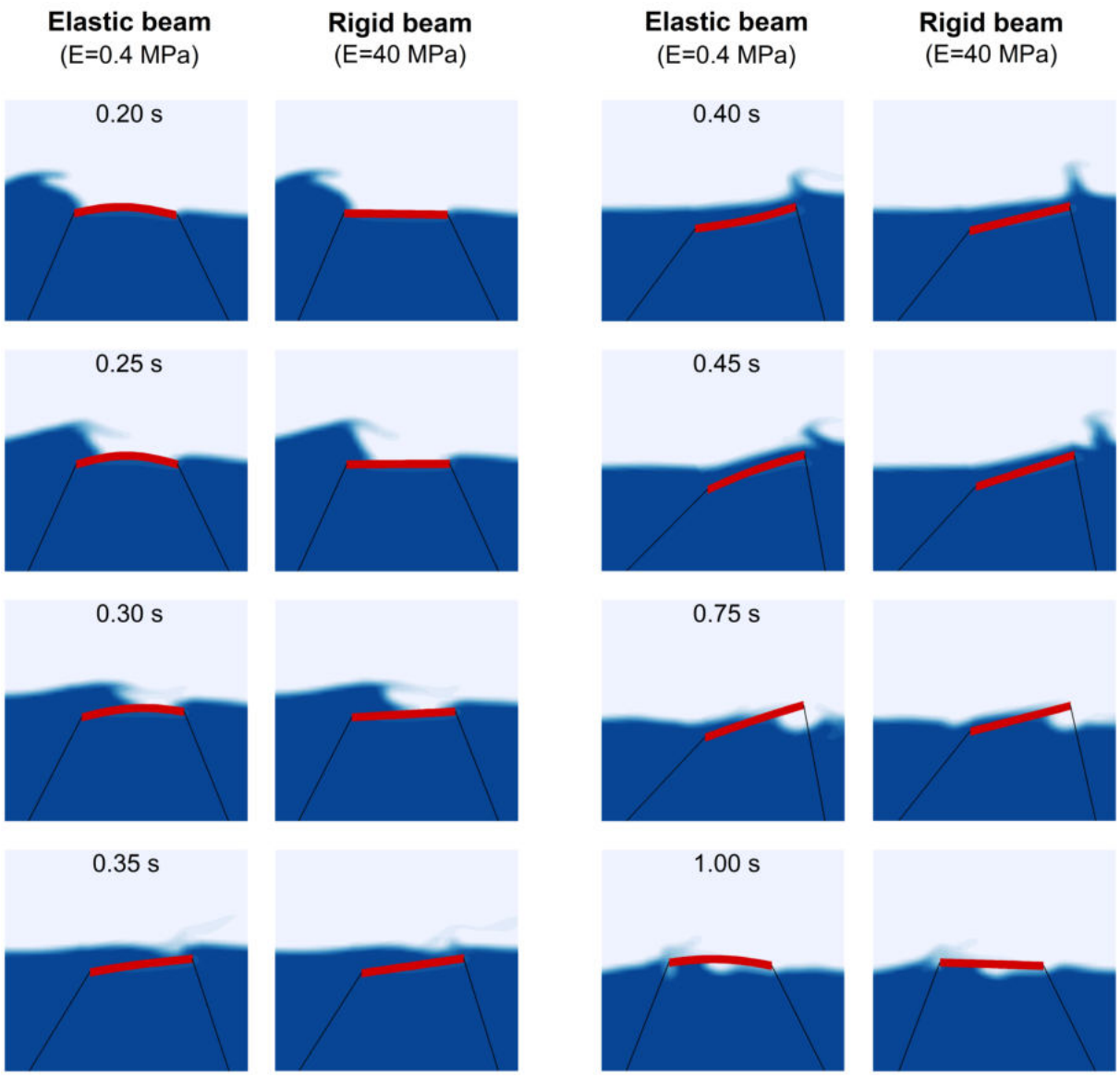


Figure 5.18.: Comparison of the structural deformation between elastic ($E = 0.4$ MPa) and rigid ($E = 40.0$ MPa) platform at different times

Figure 5.19 presents the evolution in time of the average pressure on the bottom of the platform. The flexible platform suffers from lower and less vibrating pressure than the rigid platform. The maximum average pressure for the two different Young's modules is experienced at about $t = 0.32$ s as is reported in [194]. At this time, the water wave covers 50% of the platform. The maximum pressure when $E = 0.4$ MPa is 215 Pa and when $E = 40$ MPa 269 Pa. The pressure distribution actuating below the platform at $t = 0.32$ s for each case is depicted in Figure 5.20. For the two cases, the platform's front side mainly suffers from the wave impact. The pressure on the flexible structure is almost 10% smaller than the one on the rigid structure. Figure 5.21 shows the pressure contours for the flexible platform at different times.

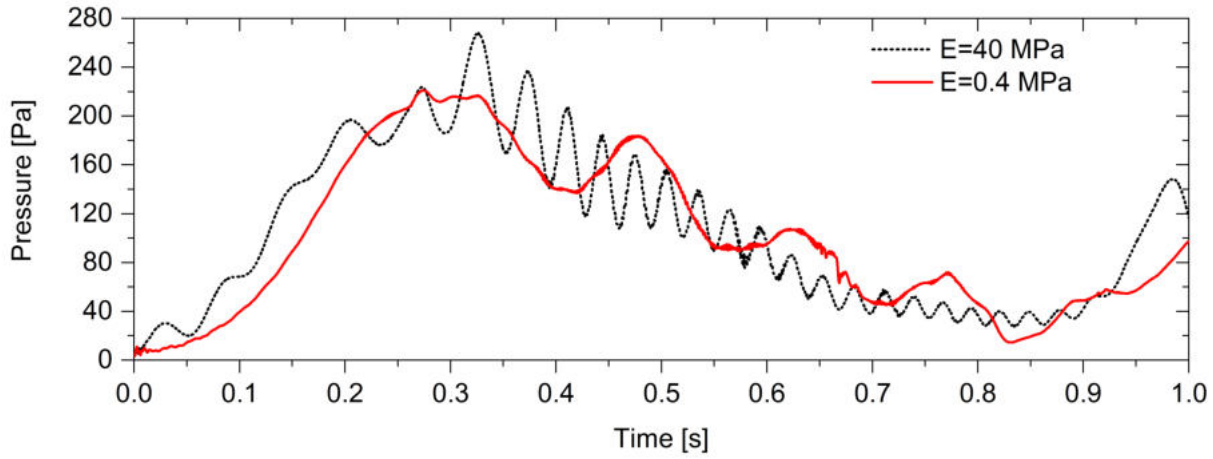


Figure 5.19.: Comparison of the structural deformation for plataforms with different Young's modules

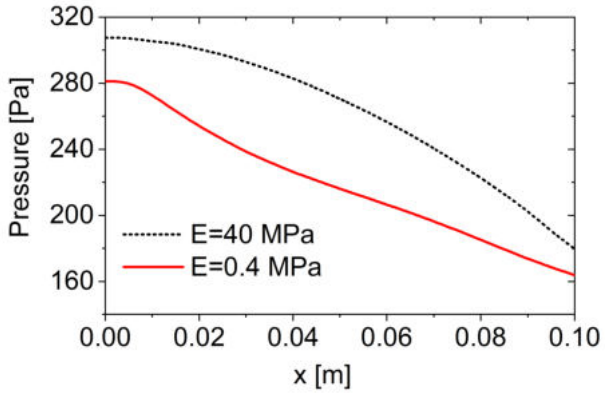


Figure 5.20.: Pressure at different positions of the plataform at $t = 0.32$ s

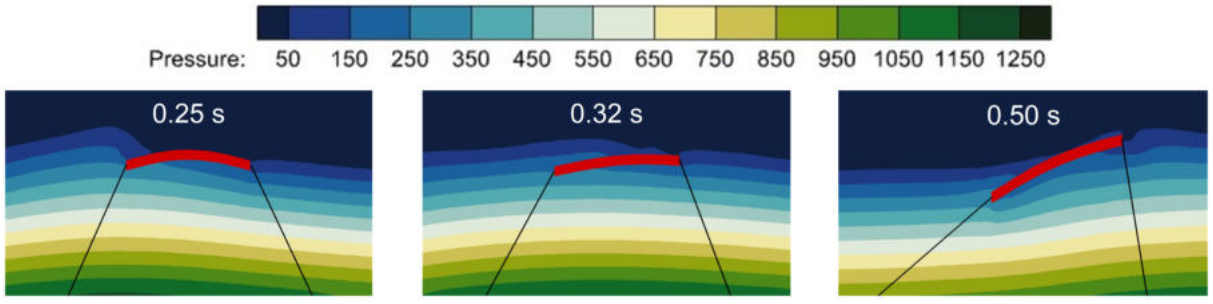


Figure 5.21.: Pressure countour plots at different times for the case with $E = 0.4$ MPa

The nonlinear deformation varies for different positions of the platform. The displacements in x and y-direction and the roll angle θ (see Figure 5.22) define the deformation of a structural node.

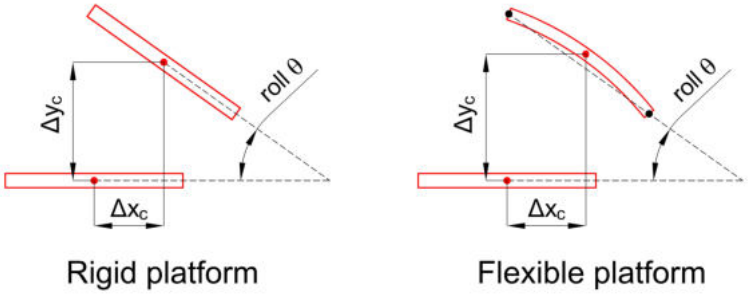


Figure 5.22.: Motion description of the platform

Figure 5.23 illustrates the roll angle of the platform for the two cases, whereas Figure 5.24 shows the x and y-displacements for the front end, center, and back end of the platforms. The figures clearly show the difference in the movement of a rigid and a flexible structure. When a rigid structure is assumed, the platform's roll angle θ and displacements change in a relatively smooth way over time. However, when taking into account a flexible structure, additional movements occur. The roll angle θ and displacements display oscillations with a higher frequency related to the platform's bending, while the overall change in the motion stays similar. This phenomenon highlights the importance of considering the flexibility of the structure in FSI when simulating the impact of water waves on floating platforms. The real flexible platform experiences more significant movements due to the impact of waves than a theoretical rigid structure. Hence, the stresses on the platform that finally determine its lifetime are more extensive than predicted by FSI simulations with rigid structures.

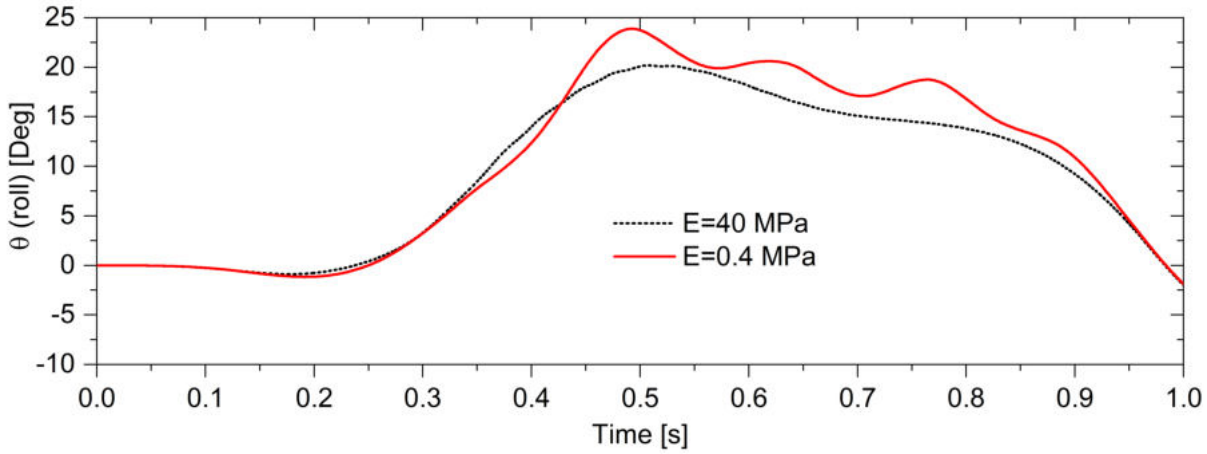


Figure 5.23.: Roll response of the platform for the two Young's modules values

Finally, regarding the computational time of these cases, it is worth mentioning that although the two simulations require about ten FSI iterations per time-step, the FSI simulation that considers the flexibility of the structure is about 60% more time expensive. The FSI computation with the rigid platform takes 0.89 hours, and the FSI computation with the flexible structure 1.44 hours.

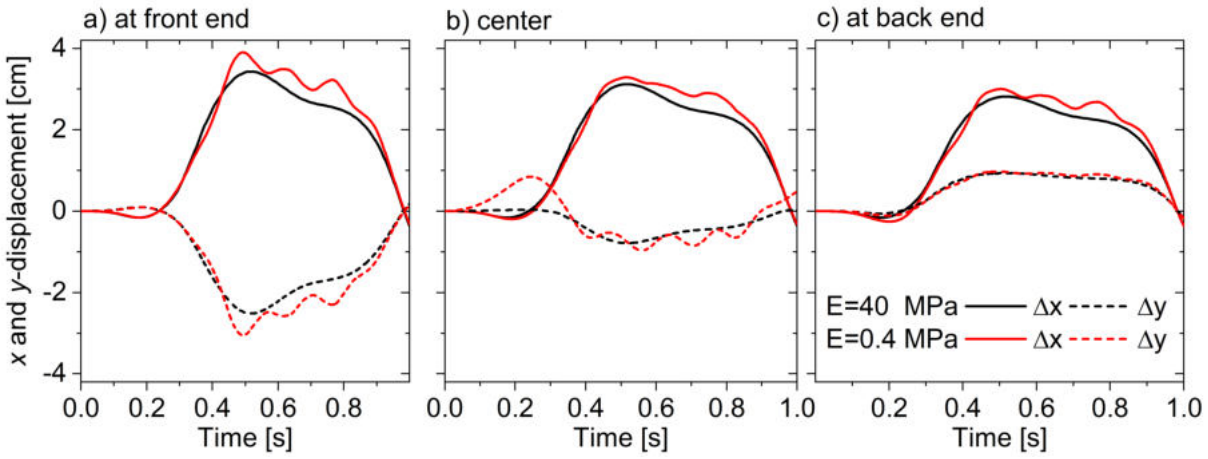


Figure 5.24.: x and y displacements response of the platform at a) front end, b) center, and c) back end.

Conclusions:

- This case highlights the importance of considering the flexibility of the structure has been demonstrated. FSI computations that only involve rigid structures can underestimate the structure's high-frequency vibration and do not predict fatigue damage.
- From the computational point of view, the proposed approach has demonstrated its applicability for more practical problems and its versatility to include more nonlinearities in the problem as the moored cables. Furthermore, the free surface profiles and structural deformation describe the physics of the problems adequately.
- With this small test case, we can also observe that FSI computations that include the structure's flexibility are much more time-consuming than those where the structure is considered rigid due to the inclusion of more nonlinearities in the problem.

6. Summary, Conclusions and Outlook

The main focus of this work was to investigate and develop numerical methodologies that can be applied to accurately and efficiently solve the dynamic of two-fluid flow, or so-called free surface flows. In a second step, the resulting methodology was integrated into a consistent partitioned fluid-structure interaction (FSI) approach suitable for predicting a flexible structure's response under high nonlinear free surface flow loads. This chapter presents a short summary, followed by some general concluding remarks about the chapters' different cross-connections. For a more detailed summary of the different topics, the reader is encouraged to review the conclusion paragraph at the final of each test case studied in this thesis. In the end, an outlook regarding further work on the improvement of the capabilities of the numerical approach is provided.

6.1. Summary and Conclusions

Considering the lack of stability and long computational time needed by current numerical FSI frameworks involving free surface flows, the question arose if enhancing the free surface flows solver could improve the accuracy, stability, and efficiency of the complete multiphysics approach, especially for the case when the two-fluids interact with flexible structures.

The new two-fluid flow solver developed in this work is based on the Finite Volume Method (FVM) that uses the in-house solver FASTEST. The originality of the present approach resides in the algorithm's stability even at high Co numbers, which allows reasonably long time-steps, reducing unnecessary computational time reported in other approaches found in the literature. The proposed two-fluid flow solver uses an algebraic Volume of Fluid (VoF) method combined with the new Modified Normalized Weighting Factor (MNWF) method developed inside the scope this research. The MNWF method numerically implements the interface capturing schemes based on an elegant treatment of the Acceptor, Donor, and Upwind volume fraction nodal values. The interface capturing scheme guarantees a sharp interface between the liquid and gas phases, and the MNWF method ensures its numerical stability even at high Courant number conditions. Also, the pressure-velocity coupling algorithm was enhanced by extra correction steps to ensure faster and better convergence of the velocity and pressure fields.

The Two-fluid flow solver was coupled with the FEM program CalculiX via the multiphysics library preCICE, resulting in a versatile partitioned framework. The free surface flow computations can be combined with different elastic theory analyses without changing the code that facilitates applying the proposed approach for more practical applications. Furthermore, the partitioned FSI can solve the fixed-point equations with an explicit or implicit coupling with only one setting in the configuration file. The solvers do not need to be modified. Also, different acceleration techniques included in the library preCICE (e.g., adaptative Aitken relaxation) can stabilize and accelerate the simulations.

The proposed two-fluid flow solver and the partitioned FSI approach have been applied to different benchmarks and validated against theoretical, experimental, and other numerical data.

The first test cases presented in Chapter 3 focused on demonstrating how a more efficient pressure-velocity coupling algorithm could drastically improve convergence rate and reduce computational time. The developed algorithm is an enhanced version of the SIMPLE algorithm that includes additional corrector PRIME steps for the pressure and velocity fields. The convergence rate increment is due to the new velocity field satisfies the momentum equation better than the one obtained by SIMPLE. The implemented code was successfully verified using the method of manufactured solutions and validated with the Drive Cavity benchmark problem.

The advantages and disadvantages of using PRIME steps were also tested. On the one hand, each additional PRIME step improves the convergence rate and numerical accuracy. It reduces the total number of iterations. On the other hand, the improvement in convergence does not directly imply a reduction in computational time. Each extra PRIME step requires re-solving the momentum and pressure-correction equations. One extra correction step has been demonstrated to be the most efficient option for working at small Courant (Co) numbers less than one. This option can, in some cases, reduce the computation time by almost 30% compared with the time used by the standard SIMPLE algorithm. For moderate Co, two or three PRIME steps are the best option to improve efficiency, while for large Co, the standard SIMPLE algorithm is the most efficient and stable option. Hence, the ideal number of extra correction PRIME steps is case-dependent.

To finalize the pressure computation, the problem with the abrupt changes of body forces near the multifluid interface, which causes parasitic currents, was treated and addressed with the solution developed by Mencinger [142]. A simple one-dimensional test was used to evaluate the effectiveness of the method. Most parasitic currents were eliminated, even though some unphysical velocity picks located at the fluid interface remain. However, these are at a level that does not affect the computations as proof of the proper resolution of the two-fluid flow validation cases. Additionally, the dependence on the relaxation factors was removed from the pressure equation to compute velocity fields independent of them.

Chapter 4 presented the general methodology to discretize the VoF transport equation in the context of the new methodology, the MNWF method, developed to deal with the implementation of interface capturing schemes. The method is based on the Normalized Weighting Factor (NWF) methodology and produces a system of tridiagonal equations. The resulting coefficient matrix is always diagonally dominant, with diagonal coefficients greater than zero, giving a numerically stable method without the necessity of under-relaxation. The main advantage of the MNWF is its high convergence rate compared to the other traditional methods. A detailed analysis of the MNWF method was published by us in [135]. Six interface capturing schemes: CICSAM, MCICSAM-W, MCICSAM-Z, HRIC, FBICS, and CUIBS (see Appendix for their formulation) were implemented with the MNWF method and used to solve two pure classical advection test cases: the slotted circle in a rotational flow and the advection of a circle in shear flow. The first case studies the volume fraction's advection in a non-uniform distribution of the Courant number, and the second the scheme's accuracy when dealing with considerable interface deformation. Whereas CICSAM, MCICSAM-W, and HRIC were excellent and accurate alternatives for calculation at low Courant number conditions, MCICSAM-Z, FBICS, and CUIBS were the more suitable schemes for working with medium-high Courant numbers. All the interface capturing schemes smeared the

interface at high Courant numbers ($Co > 0.8$), which causes a deterioration of the sharpness or nonphysical changes of the flow interface.

Our main objective was to develop a suitable two-fluid flow solver for use in FSI applications. Consequently, only the interface capturing schemes that better work at medium-high Co numbers, MCICSAM-Z, FBICS, and CUIBS, were considered to validate the solver. Four test cases validated the solver and demonstrated the accuracy and efficiency of the methodology. The topics covered in the validation were the 3D sloshing phenomenon, the 3D bubbly dynamics with surface tension dominant effects, waves capture capabilities in the Rayleigh-Taylor instabilities, and the break-water-dam impacting a rigid structure. The results were in excellent agreement with the analytical, numerical, and experimental references results. For these cases, the most accurate schemes were CUIBS and FBICS. They are low numerical diffusive interface capturing schemes and allow using a large time-step without losing stability. Hence, they were found to be good candidates for capturing flow interfaces in FSI applications.

The second meaningful contribution of this work has been the redesign of the FSI framework that couples the in-house solver FASTEST with the structural solver CalculiX via preCICE. The novel FSI coupling algorithm is a versatile framework that allows the simulation of diverse applications as the FSI3 benchmark case used to validate it. The displacement and forces matched the reference results.

The proposed numerical approach was used to solve free surface flows interacting with flexible structures. The accuracy of the method was checked by comparing the obtained results with numerical and experimental reference data. The cases conducted were the dam-break water interacting with an elastic obstruction, an elastic plate, and a moored deformable structure. The results present an overall good agreement with the reference data. The violent free surface flows and large structural deformations were addressed with reasonably good accuracy and efficiency. For the first two applications, the phenomena' main features at the initial impacting stage were adequately captured by the numerical approach. The main differences observed in the present work have been attributed to the lack of treatment of the highly nonlinear turbulences present in these cases and the experiments' intrinsic complexity (Section 5.6.2). The last application was a more practical engineer application aimed to demonstrate the influence of the structure's flexibility in the deformation response. The results revealed that assuming a rigid structure could underestimate the structure's high-frequency vibration and not predict fatigue damage.

Finally, close attention has been paid to the computational efficiency of the general FSI framework. It has been shown that the present method is fast and can obtain good results in a couple of hours, especially during the simulation of rigid structures. Therefore, the novel FSI approach can be considered an attractive candidate for the simulation of wave-structure interaction problems typically found in the ocean and coastal engineering disciplines.

6.2. Outlook

As shown in break-dam and elastic structure problems, the current FSI approach involving free surface flows can reproduce the phenomenon with acceptably good accuracy in a reasonable time. However, the violent turbulence flow reported in the experiments cannot be predicted precisely by the present numerical approach. Therefore, future work will focus on integrating turbulence models in the two-fluid flow formulations to study appropriately violent wave impacts against

flexible bodies, such as those occurring in sloshing tanks.

Moreover, for solving more complex problems in the ocean, coastal and naval engineering in the future, it would be useful to include a wave generator and a damping model in the FSI approach. The first can produce regular and large waves with realistic frequencies that can be used for a better prediction of the structure deformations. The second avoids the wave reflection effect without enlarging the computational domain, thus reducing calculation time.

Bibliography

[1] Adelsberger, J. et al. “3D incompressible two-phase flow benchmark computations for rising droplets”. In: *The 11th world congress on computational mechanics (WCCM XI)*. July. Barcelona, Spain, 2014, Vol. 179.

[2] Aghajari, N. and Schäfer, M. “Efficient shape optimization for fluid–structure interaction problems”. In: *Journal of Fluids and Structures* 57 (Aug. 2015), pp. 298–313.

[3] Akkerman, I. et al. “Free-Surface Flow and Fluid-Object Interaction Modeling With Emphasis on Ship Hydrodynamics”. In: *Journal of Applied Mechanics* 79.1 (Jan. 2012), p. 010905.

[4] Anderson, J. D. *Computational Fluid Dynamics: the Basics with Applications*. New York: McGraw-Hill Science., 1995, p. 547.

[5] Antoci, C., Gallati, M., and Sibilla, S. “Numerical simulation of fluid–structure interaction by SPH”. In: *Computers & Structures* 85.11-14 (June 2007), pp. 879–890.

[6] Ashgriz, N. and Poo, J. “FLAIR: Flux line-segment model for advection and interface reconstruction”. In: *Journal of Computational Physics* 93.2 (Apr. 1991), pp. 449–468.

[7] Bai, X. et al. “Numerical simulation of a marine current turbine in free surface flow”. In: *Renewable Energy* 63 (Mar. 2014), pp. 715–723.

[8] Banks, J., Aslam, T., and Rider, W. “On sub-linear convergence for linearly degenerate waves in capturing schemes”. In: *Journal of Computational Physics* 227.14 (July 2008), pp. 6985–7002.

[9] Barone, M. and Payne, J. “Methods for simulation-based analysis of fluid-structure interaction”. In: *Sandia Report* 6573.October (2005), p. 10.

[10] Basting, S. and Weismann, M. “A hybrid level set–front tracking finite element approach for fluid–structure interaction and two-phase flow applications”. In: *Journal of Computational Physics* 255 (Dec. 2013), pp. 228–244.

[11] Bathe, K.-J. *Finite element procedures*. Second. United States of America, 2014, p. 1043.

[12] Bathe, K.-J. and Zhang, H. “Finite element developments for general fluid flows with structural interactions”. In: *International Journal for Numerical Methods in Engineering* 60.1 (May 2004), pp. 213–232.

[13] Bazilevs, Y. et al. “Simulating Free-Surface FSI and Fatigue Damage in Wind-Turbine Structural Systems”. In: *Frontiers in Computational Fluid-Structure Interaction and Flow Simulation: Research from Lead Investigators under Forty – 2018*. Ed. by T. E. Tezduyar. Cham: Springer International Publishing, 2018, pp. 1–28.

[14] Bazilevs, Y. et al. “Isogeometric Fluid–structure Interaction Analysis with Applications to Arterial Blood Flow”. In: *Computational Mechanics* 38.4-5 (Sept. 2006), pp. 310–322.

[15] Bogaers, A. et al. “An evaluation of quasi-Newton methods for application to FSI problems involving free surface flow and solid body contact”. In: *Computers & Structures* 173 (Sept. 2016), pp. 71–83.

[16] Bogaers, A. et al. “Quasi-Newton methods for implicit black-box FSI coupling”. In: *Computer Methods in Applied Mechanics and Engineering* 279 (2014), pp. 113–132.

[17] Brackbill, J., Kothe, D., and Zemach, C. “A continuum method for modeling surface tension”. In: *Journal of Computational Physics* 100.2 (June 1992), pp. 335–354.

[18] Bungartz, H.-J. et al. “A plug-and-play coupling approach for parallel multi-field simulations”. In: *Computational Mechanics* 55.6 (2015), pp. 1119–1129.

[19] Bungartz, H.-J. et al. “preCICE – A fully parallel library for multi-physics surface coupling”. In: *Computers & Fluids* 141 (Dec. 2016), pp. 250–258.

[20] Calderer, A. et al. “Fluid–structure interaction simulation of floating structures interacting with complex, large-scale ocean waves and atmospheric turbulence with application to floating offshore wind turbines”. In: *Journal of Computational Physics* 355 (Feb. 2018), pp. 144–175.

[21] Causin, P., Gerbeau, J. F., and Nobile, F. “Added-mass effect in the design of partitioned algorithms for fluid-structure problems”. In: *Computer Methods in Applied Mechanics and Engineering* 194.42-44 (2005), pp. 4506–4527.

[22] Cebral, J. R. and Lohner, R. “Conservative Load Projection and Tracking for Fluid-Structure Problems”. In: *AIAA Journal* 35.4 (Apr. 1997), pp. 687–692.

[23] Ceniceros, H. D., Nós, R. L., and Roma, A. M. “Three-dimensional, fully adaptive simulations of phase-field fluid models”. In: *Journal of Computational Physics* 229.17 (Aug. 2010), pp. 6135–6155.

[24] Chan, R. K.-C. and Street, R. L. “A computer study of finite-amplitude water waves”. In: *Journal of Computational Physics* 6.1 (Aug. 1970), pp. 68–94.

[25] Chandrasekhar, S. *Hydrodynamic and hydromagnetic stability*. Courier Corporation, 2013.

[26] Chen, X., Schäfer, M., and Bothe, D. “Numerical modeling and investigation of viscoelastic fluid–structure interaction applying an implicit partitioned coupling algorithm”. In: *Journal of Fluids and Structures* 54 (Apr. 2015), pp. 390–421.

[27] Choudhary, A. et al. “Code verification for multiphase flows using the method of manufactured solutions”. In: *International Journal of Multiphase Flow* 80 (Apr. 2016), pp. 150–163.

[28] Chourushi, T. “Computationally inexpensive and revised normalized weighting factor method for segregated solvers”. In: *International Journal of Computer Mathematics* 95.8 (Aug. 2018), pp. 1622–1653.

[29] Ciortan, C., Wanderley, J., and Guedes Soares, C. “Free surface flow around a ship model using an interface-capturing method”. In: *Ocean Engineering* 44 (Apr. 2012), pp. 57–67.

[30] Coward, A. V. et al. “Temporal evolution of periodic disturbances in two-layer Couette flow”. In: *Journal of Computational Physics* 132.2 (1997), pp. 346–361.

[31] Daly, B. J. “Numerical Study of Two Fluid Rayleigh-Taylor Instability”. In: *Physics of Fluids* 10.2 (1967), p. 297.

[32] Darwish, M. and Moukalled, F. “Convective Schemes for Capturing Interfaces of Free-Surface Flows on Unstructured Grids”. In: *Numerical Heat Transfer, Part B: Fundamentals* 49.1 (Jan. 2006), pp. 19–42.

[33] Darwish, M. and Moukalled, F. “Normalized variable and space formulation methodology for high-resolution schemes”. In: *Numerical Heat Transfer, Part B: Fundamentals* 26.1 (July 1994), pp. 79–96.

[34] Darwish, M. and Moukalled, F. “The normalized weighting factor method: a novel technique for accelerating the convergence of high-resolution convective schemes”. In: *Numerical Heat Transfer, Part B: Fundamentals* 30.2 (Sept. 1996), pp. 217–237.

[35] Davis, S. F. “Flux difference splittings and limiters for the resolution of contact discontinuities”. In: *Applied Mathematics and Computation* 65.1-3 (Sept. 1994), pp. 3–18.

[36] De Boer, A., Van Zuijlen, A., and Bijl, H. “Review of coupling methods for non-matching meshes”. In: *Computer Methods in Applied Mechanics and Engineering* 196.8 (Jan. 2007), pp. 1515–1525.

[37] De Boer, A., Van Zuijlen, A., and Bijl, H. “Comparison of conservative and consistent approaches for the coupling of non-matching meshes”. In: *Computer Methods in Applied Mechanics and Engineering* 197.49-50 (Sept. 2008), pp. 4284–4297.

[38] Degroote, J. “Partitioned Simulation of Fluid-Structure Interaction”. In: *Archives of Computational Methods in Engineering* 20.3 (Sept. 2013), pp. 185–238.

[39] Degroote, J., Bathe, K.-J., and Vierendeels, J. “Performance of a new partitioned procedure versus a monolithic procedure in fluid-structure interaction”. In: *Computers and Structures* 87.11-12 (2009), pp. 793–801.

[40] Degroote, J. et al. “Stability of a coupling technique for partitioned solvers in FSI applications”. In: *Computers & Structures* 86.23-24 (Dec. 2008), pp. 2224–2234.

[41] Demirdžić, I. and Perić, M. “Space conservation law in finite volume calculations of fluid flow”. In: *International Journal for Numerical Methods in Fluids* 8.9 (Sept. 1988), pp. 1037–1050.

[42] Dettmer, W. G. and Perić, D. “A Fully Implicit Computational Strategy for Strongly Coupled Fluid–Solid Interaction”. In: *Archives of Computational Methods in Engineering* 14.3 (Aug. 2007), pp. 205–247.

[43] Dhondt, G. “CalculiX CrunchiX USER ’ S MANUAL”. In: (2012), p. 2012.

[44] Dhondt, G. *The Finite Element Method for Three-Dimensional Thermomechanical Applications*. Chichester, UK: John Wiley & Sons, Ltd, May 2004.

[45] Donea, J. et al. “Arbitrary Lagrangian-Eulerian Methods”. In: *Encyclopedia of Computational Mechanics Second Edition*. Chichester, UK: John Wiley & Sons, Ltd, Dec. 2017, pp. 1–23.

[46] Durst, F. and Schäfer, M. “A parallel block-structured multigrid method for the prediction of incompressible flows”. In: *International Journal for Numerical Methods in Fluids* 22.6 (Mar. 1996), pp. 549–565.

[47] Efremov, V. R. et al. “Method for taking into account gravity in free-surface flow simulation”. In: *Computational Mathematics and Mathematical Physics* 57.10 (Oct. 2017), pp. 1720–1733.

[48] Falahaty, H., Khayyer, A., and Gotoh, H. “Enhanced particle method with stress point integration for simulation of incompressible fluid-nonlinear elastic structure interaction”. In: *Journal of Fluids and Structures* 81 (Aug. 2018), pp. 325–360.

[49] Fang, H. R. and Saad, Y. “Two classes of multisecant methods for nonlinear acceleration”. In: *Numerical Linear Algebra with Applications* 16.3 (2009), pp. 197–221.

[50] Farhat, C. and Lesoinne, M. “Two efficient staggered algorithms for the serial and parallel solution of three-dimensional nonlinear transient aeroelastic problems”. In: *Computer Methods in Applied Mechanics and Engineering* 182.3-4 (Feb. 2000), pp. 499–515.

[51] Farhat, C., Lesoinne, M., and Le Tallec, P. “Load and motion transfer algorithms for fluid/structure interaction problems with non-matching discrete interfaces: Momentum and energy conservation, optimal discretization and application to aeroelasticity”. In: *Computer Methods in Applied Mechanics and Engineering* 157.1-2 (Apr. 1998), pp. 95–114.

[52] Farhat, C., Geuzaine, P., and Brown, G. “Application of a three-field nonlinear fluid–structure formulation to the prediction of the aeroelastic parameters of an F-16 fighter”. In: *Computers & Fluids* 32.1 (Jan. 2003), pp. 3–29.

[53] Farhat, C., Geuzaine, P., and Grandmont, C. “The discrete geometric conservation law and the nonlinear stability of ALE schemes for the solution of flow problems on moving grids”. In: *Journal of Computational Physics* 174.2 (2001), pp. 669–694.

[54] Fekken, G. “Numerical simulation of free-surface flow with moving rigid bodies”. Doctoral. Groningen: s.n., 2004.

[55] Felippa, C. A., Park, K., and Farhat, C. “Partitioned analysis of coupled mechanical systems”. In: *Computer Methods in Applied Mechanics and Engineering* 190.24-25 (Mar. 2001), pp. 3247–3270.

[56] Fernández, M. A. “Coupling schemes for incompressible fluid-structure interaction: implicit, semi-implicit and explicit”. In: *SeMA Journal* 55.1 (2011), pp. 59–108.

[57] Fernández, M. A., Gerbeau, J.-F., and Grandmont, C. “A projection algorithm for fluid–structure interaction problems with strong added-mass effect”. In: *Comptes Rendus Mathematique* 342.4 (Feb. 2006), pp. 279–284.

[58] Fernández, M. A. and Moubachir, M. “A Newton method using exact jacobians for solving fluid–structure coupling”. In: *Computers & Structures* 83.2-3 (Jan. 2005), pp. 127–142.

[59] Ferziger, J. H. and Perić, M. “Methods for Unsteady Problems”. In: *Computational Methods for Fluid Dynamics*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1996, pp. 127–148.

[60] Ferziger, J. H., Perić, M., and Street, R. L. *Computational Methods for Fluid Dynamics*. Cham: Springer International Publishing, 2020.

[61] Fletcher, C. *Computational Techniques for Fluid Dynamics 2: Specific techniques for different flow categories.* Springer Science & Business Media, 2012.

[62] Formaggia, L. and Nobile, F. “Stability analysis of second-order time accurate schemes for ALE–FEM”. In: *Computer Methods in Applied Mechanics and Engineering* 193.39-41 (Oct. 2004), pp. 4097–4116.

[63] Friedrich, J. and Schäfer, M. “Towards an Acoustic Simulation of a Water Drop Impacting in a Water Pool”. In: *Flow, Turbulence and Combustion* (Apr. 2020).

[64] Fung, Y. *Foundations of solid mechanics*. New Jersey: Prentice-Hall, 1965, p. 525.

[65] Garelli, L. et al. “Geometric conservation law in ALE formulations”. In: *Mecánica Computacional* XXVIII (2009), pp. 1477–1488.

[66] Gaskell, P. H. and Lau, A. K. C. “Curvature-compensated convective transport: SMART, A new boundedness- preserving transport algorithm”. In: *International Journal for Numerical Methods in Fluids* 8.6 (June 1988), pp. 617–641.

[67] Gatzhammer, B. “Efficient and Flexible Partitioned Simulation of Fluid-Structure Interactions”. Ph.D. Thesis. Technische Universität München, 2014.

[68] Gillebaart, T. et al. “Time consistent fluid structure interaction on collocated grids for incompressible flow”. In: *Computer Methods in Applied Mechanics and Engineering* 298 (Jan. 2016), pp. 159–182.

[69] Gingold, R. A. and Monaghan, J. J. “Smoothed particle hydrodynamics: theory and application to non-spherical stars”. In: *Monthly Notices of the Royal Astronomical Society* 181.3 (Dec. 1977), pp. 375–389.

[70] Glowinski, R. et al. “A distributed Lagrange multiplier/fictitious domain method for particulate flows”. In: *International Journal of Multiphase Flow* 25.5 (Aug. 1999), pp. 755–794.

[71] Glück, M. et al. “Computation of fluid–structure interaction on lightweight structures”. In: *Journal of Wind Engineering and Industrial Aerodynamics* 89.14-15 (Dec. 2001), pp. 1351–1368.

[72] Gnesin, V., Rządkowski, R., and Kolodyazhnaya, L. “A Coupled Fluid-Structure Analysis for 3D Flutter in Turbomachines”. In: *Volume 4: Manufacturing Materials and Metallurgy; Ceramics; Structures and Dynamics; Controls, Diagnostics and Instrumentation; Education*. American Society of Mechanical Engineers, May 2000.

[73] Gomes, J. P. et al. “Experimental and numerical study on a laminar fluid–structure interaction reference test case”. In: *Journal of Fluids and Structures* 27.1 (Jan. 2011), pp. 43–61.

[74] Gomez, H. and Bueno, J. *Frontiers in Computational Fluid-Structure Interaction and Flow Simulation*. Ed. by T. E. Tezduyar. Modeling and Simulation in Science, Engineering and Technology. Cham: Springer International Publishing, 2018, pp. 131–165.

[75] Gong, K. et al. “Two-phase SPH simulation of fluid–structure interactions”. In: *Journal of Fluids and Structures* 65 (Aug. 2016), pp. 155–179.

[76] Gottfried, D. A. and Fleeter, S. “Aerodynamic Damping Predictions in Turbomachines Using a Coupled Fluid-Structure Model”. In: *Journal of Propulsion and Power* 21.2 (Mar. 2005), pp. 327–334.

[77] Gu, H. et al. “Development of a free surface flow solver for the simulation of wave/body interactions”. In: *European Journal of Mechanics - B/Fluids* 38 (Mar. 2013), pp. 1–17.

[78] Haelterman, R. et al. “Improving the performance of the partitioned QN-ILS procedure for fluid–structure interaction problems: Filtering”. In: *Computers & Structures* 171 (July 2016), pp. 9–17.

[79] Harlow, F. H. and Welch, J. E. “Numerical Calculation of Time-Dependent Viscous Incompressible Flow of Fluid with Free Surface”. In: *Physics of Fluids* 8.12 (Mar. 1965), p. 2182.

[80] Heil, M. “An efficient solver for the fully coupled solution of large-displacement fluid–structure interaction problems”. In: *Computer Methods in Applied Mechanics and Engineering* 193.1-2 (Jan. 2004), pp. 1–23.

[81] Hino, T. “CFD-based estimation of propulsive performance in ship design”. In: *26th symposium on naval hydrodynamics*. Rome, 2006.

[82] Hirt, C. and Nichols, B. “Volume of fluid (VOF) method for the dynamics of free boundaries”. In: *Journal of Computational Physics* 39.1 (Jan. 1981), pp. 201–225.

[83] Hoekstra, M. et al. “Free-surface flow modelling with interface capturing techniques”. In: *MARINE 2007*. Vol. 2. Barcelona: International Conference on Computational Methods in Marine Engineering, 2007, pp. 1–4.

[84] Hoekstra, M. “PARNASSOS: An efficient method for ship stern flow calculation”. In: *Third Osaka Colloquium on Advanced CFD Applications to Ship Flow and Hull Form Design*. 1998, pp. 331–357.

[85] Hogg, P. W., Gu, X.-J., and Emerson, D. R. “An implicit algorithm for capturing sharp fluid interfaces in the volume of fluid advection method”. In: *European Conference on Computational Fluid Dynamics ECCOMAS CFD*. 2006.

[86] Holzmann, T. *Mathematics, numerics, derivations and openfoam*. Release 7. Holzmann CFD, 2020.

[87] Hou, G., Wang, J., and Layton, A. “Numerical Methods for Fluid-Structure Interaction — A Review”. In: *Communications in Computational Physics* 12.2 (Aug. 2012), pp. 337–377.

[88] *http://www.featflow.de*.

[89] *https://openfoam.org*.

[90] Hu, Z. et al. “Numerical study of rogue wave overtopping with a fully-coupled fluid-structure interaction model”. In: *Ocean Engineering* 137 (June 2017), pp. 48–58.

[91] Idelsohn, S. et al. “Unified Lagrangian formulation for elastic solids and incompressible fluids: Application to fluid–structure interaction problems via the PFEM”. In: *Computer Methods in Applied Mechanics and Engineering* 197.19-20 (Mar. 2008), pp. 1762–1776.

[92] Irons, B. M. and Tuck, R. C. “A version of the Aitken accelerator for computer iteration”. In: *International Journal for Numerical Methods in Engineering* 1.3 (July 1969), pp. 275–277.

[93] Issa, R. “Solution of the implicitly discretised fluid flow equations by operator-splitting”. In: *Journal of Computational Physics* 62.1 (Jan. 1986), pp. 40–65.

[94] James, I. “A high-performance explicit vertical advection scheme for ocean models: how PPM can beat the CFL condition”. In: *Applied Mathematical Modelling* 24.1 (Jan. 2000), pp. 1–9.

[95] Jasak, H., Weller, H., and Gosman, A. “High resolution NVD differencing scheme for arbitrarily unstructured meshes”. In: *International Journal for Numerical Methods in Fluids* 31.2 (Sept. 1999), pp. 431–449.

[96] Jasak, H. and Weller, H. *Interface Tracking Capabilities of the Inter-Gamma Differencing Scheme*. Tech. rep. Department of Mechanical Engineering;Imperial College of Science; Technology and Medicine., 1995.

[97] Jhung, M. J. “Assessment of thermal fatigue in mixing tee by FSI analysis”. In: *Nuclear Engineering and Technology* 45.1 (Feb. 2013), pp. 99–106.

[98] Kadja, M., Anagnostopoulos, J. S., and Bergeles, G. C. “Study of wind flow and pollutant dispersion by newly developed precision-improving methods”. In: *International communications in heat and mass Transfer* 23.8 (1996), pp. 1065–1076.

[99] Kang, S. and Sotiropoulos, F. “Numerical modeling of 3D turbulent free surface flow in natural waterways”. In: *Advances in Water Resources* 40 (May 2012), pp. 23–36.

[100] Karim, M. M., Prasad, B., and Rahman, N. “Numerical simulation of free surface water wave for the flow around NACA 0015 hydrofoil using the volume of fluid (VOF) method”. In: *Ocean Engineering* 78 (Mar. 2014), pp. 89–94.

[101] Kassiotis, C., Ibrahimbegovic, A., and Matthies, H. “Partitioned solution to fluid–structure interaction problem in application to free-surface flows”. In: *European Journal of Mechanics - B/Fluids* 29.6 (Nov. 2010), pp. 510–521.

[102] Kawaguchi, Y., Tao, W.-Q., and Ozoe, H. “Checkerboard pressure predictions due to the underrelaxation factor and time step size for a nonstaggered grid with momentum interpolation method”. In: *Numerical Heat Transfer, Part B: Fundamentals* 41.1 (Jan. 2002), pp. 85–94.

[103] Khan, M. U. et al. “Multiphysics out of position airbag simulation”. In: *International Journal of Crashworthiness* 13.2 (Mar. 2008), pp. 159–166.

[104] Kim, G.-H. and Park, S. “Development of a numerical simulation tool for efficient and robust prediction of ship resistance”. In: *International Journal of Naval Architecture and Ocean Engineering* 9.5 (Sept. 2017), pp. 537–551.

[105] Kim, N.-H. *Introduction Analysis Finite Element to Nonlinear*. 2018.

[106] Knupp, P. and Salari, K. *Verification of computer codes in computational science and engineering*. Ed. by K. H. Rosen. Chapman and Hall/CRC, 2002.

[107] Kölke, A. et al. “Strongly Coupled Analysis of Fluid-Structure Interaction with Free Surface Flow”. In: *PAMM* 4.1 (Dec. 2004), pp. 338–339.

[108] Koshizuka, S. “A particle method for incompressible viscous flow with fluid fragmentation”. In: *Comput. Fluid Dyn. J.* 4.29 (1995).

[109] Koshizuka, S. and Oka, Y. “Moving-Particle Semi-Implicit Method for Fragmentation of Incompressible Fluid”. In: *Nuclear Science and Engineering* 123.3 (July 1996), pp. 421–434.

[110] Kothe, D. B. “Perspective on Eulerian Finite Volume Methods for Incompressible Interfacial Flows”. In: *Free Surface Flows*. Vol. M. Vienna: Springer Vienna, 1998, pp. 267–331.

[111] Küttler, U. and Wall, W. A. “Fixed-point fluid–structure interaction solvers with dynamic relaxation”. In: *Computational Mechanics* 43.1 (Dec. 2008), pp. 61–72.

[112] Lapidus, L. and Pinder, G. *Numerical solution of partial differential equations in science and engineering*. New York: John Wiley & Sons, 1982.

[113] Lee, C.-S. et al. “Evaluation of sloshing resistance performance for LNG carrier insulation system based on fluid-structure interaction analysis”. In: *International Journal of Naval Architecture and Ocean Engineering* 5.1 (Mar. 2013). Ed. by C. Quan et al., pp. 1–20.

[114] Lehnhäuser, T. and Schäfer, M. “Efficient discretization of pressure-correction equations on non-orthogonal grids”. In: *International Journal for Numerical Methods in Fluids* 42.2 (2003), pp. 211–231.

[115] Lehnhäuser, T. and Schäfer, M. “Improved linear interpolation practice for finite-volume schemes on complex grids”. In: *International Journal for Numerical Methods in Fluids* 38.7 (Mar. 2002), pp. 625–645.

[116] Leonard, B. “A stable and accurate convective modelling procedure based on quadratic upstream interpolation”. In: *Computer Methods in Applied Mechanics and Engineering* 19.1 (June 1979), pp. 59–98.

[117] Leonard, B. *Sharp simulation of discontinuities in highly convective steady flow*. Tech. rep. NASA-TM-100240, 1987.

[118] Leonard, B. “The ULTIMATE conservative difference scheme applied to unsteady one-dimensional advection”. In: *Computer Methods in Applied Mechanics and Engineering* 88.1 (June 1991), pp. 17–74.

[119] Leonard, B. *Universal limiter for transient interpolation modeling of the advective transport equations: the ULTIMATE conservative difference*. Tech. rep. NASA TR-100916, ICOMP-88-11, 1988.

[120] Leonard, B. and Mokhtari, S. “Beyond first-order upwinding: The ultra-sharp alternative for non-oscillatory steady-state simulation of convection”. In: *International Journal for Numerical Methods in Engineering* 30.4 (Sept. 1990), pp. 729–766.

[121] Lian, Y., Zhang, X., and Liu, Y. “Coupling of finite element method with material point method by local multi-mesh contact method”. In: *Computer Methods in Applied Mechanics and Engineering* 200.47-48 (Nov. 2011), pp. 3482–3494.

[122] Liao, K. and Hu, C. “A coupled FDM–FEM method for free surface flow interaction with thin elastic plate”. In: *Journal of Marine Science and Technology* 18.1 (Mar. 2013), pp. 1–11.

[123] Liao, K., Hu, C., and Sueyoshi, M. “Free surface flow impacting on an elastic structure: Experiment versus numerical simulation”. In: *Applied Ocean Research* 50 (Mar. 2015), pp. 192–208.

[124] Lien, F. and Leschziner, M. “A general non-orthogonal collocated finite volume algorithm for turbulent flow at all speeds incorporating second-moment turbulence-transport closure, Part 1: Computational implementation”. In: *Computer Methods in Applied Mechanics and Engineering* 114.1-2 (Apr. 1994), pp. 123–148.

[125] Lin, S.-Y. et al. “A parallel VOF IB pressure-correction method for simulation of multiphase flows”. In: *Applied Mathematical Modelling* 40.3 (Feb. 2016), pp. 1800–1815.

[126] Lindner, F. et al. “A comparison of various quasi-Newton schemes for partitioned fluid-structure interaction”. In: *6th International Conference on Coupled Problems in Science and Engineering*. 2015, pp. 477–488.

[127] Link, G. et al. “A 2D finite-element scheme for fluid–solid–acoustic interactions and its application to human phonation”. In: *Computer Methods in Applied Mechanics and Engineering* 198.41-44 (Sept. 2009), pp. 3321–3334.

[128] Liu, J., Jaiman, R. K., and Gurugubelli, P. S. “A stable second-order scheme for fluid–structure interaction with strong added-mass effects”. In: *Journal of Computational Physics* 270 (Aug. 2014), pp. 687–710.

[129] Liu, M., Shao, J., and Li, H. “An SPH model for free surface flows with moving rigid objects”. In: *International Journal for Numerical Methods in Fluids* 74.9 (Mar. 2014), pp. 684–697.

[130] Lötstedt, P. “A front tracking method applied to Burgers’ equation and two-phase porous flow”. In: *Journal of Computational Physics* 47.2 (Aug. 1982), pp. 211–228.

[131] Ma, R. et al. “On the Geometric Conservation Law for Unsteady Flow Simulations on Moving Mesh”. In: *Procedia Engineering* 126 (2015), pp. 639–644.

[132] Majumdar, S. “Role of underrelaxation in momentum interpolation for calculation of flow with nonstaggered grids”. In: *Numerical Heat Transfer* 13.1 (Jan. 1988), pp. 125–132.

[133] Maliska, C. and Raithby, G. “Calculating three-dimensional fluid flows using nonorthogonal grids”. In: *Third international conference on numerical methods in laminar and turbulent flows*. Seattle, 1983, pp. 656–666.

[134] Mariño, J. and Schäfer, M. “Investigation of discretization methods for simulating multiphase flows with moving grids”. In: *10th International Conference on Multiphase Flow*. Rio de Janeiro, 2019.

[135] Mariño-Salguero, J. and Schäfer, M. “A Modified Normalized Weighting Factor method for improving the efficiency of the blended high-resolution advection schemes in the context of multiphase flows”. In: *Experimental and Computational Multiphase Flow* 2.4 (July 2020).

[136] Marschall, H. “Towards the Numerical Simulation of Multi-Scale Two-Phase Flows”. Doctoral. Technischen Universität München, 2011.

[137] Martínez-Ferrer, P. J. et al. “An efficient finite-volume method to study the interaction of two-phase fluid flows with elastic structures”. In: *Journal of Fluids and Structures* 83 (Nov. 2018), pp. 54–71.

[138] Masud, A. “Effects of Mesh Motion on the Stability and Convergence of ALE Based Formulations for Moving Boundary Flows”. In: *Computational Mechanics* 38.4-5 (Sept. 2006), pp. 430–439.

[139] Matthies, H. G. and Steindorf, J. “Partitioned strong coupling algorithms for fluid–structure interaction”. In: *Computers & Structures* 81.8-11 (May 2003), pp. 805–812.

[140] Mehl, M. et al. “Parallel coupling numerics for partitioned fluid–structure interaction simulations”. In: *Computers & Mathematics with Applications* 71.4 (Feb. 2016), pp. 869–891.

[141] Ménard, T., Tanguy, S., and Berlemont, A. “Coupling level set/VOF/ghost fluid methods: Validation and application to 3D simulation of the primary break-up of a liquid jet”. In: *International Journal of Multiphase Flow* 33.5 (May 2007), pp. 510–524.

[142] Mencinger, J. “An Alternative Finite Volume Discretization of Body Force Field on Collocated Grid”. In: *Finite Volume Method - Powerful Means of Engineering Design*. Vol. Di. March 2012. InTech, Mar. 2012.

[143] Mencinger, J. and Žun, I. “On the finite volume discretization of discontinuous body force field on collocated grid: Application to VOF method”. In: *Journal of Computational Physics* 221.2 (Feb. 2007), pp. 524–538.

[144] Meyer, J. et al. “Advanced CDF-Simulations of free-surface flows around modern sailing yachts using a newly developed OpenFOAM solver”. In: *The 22nd Chesapeake sailing yacht symposium*. The 22nd Chesapeake Sailing Yacht Symposium. ANNAPOLIS, 2016.

[145] Ming, P. et al. “A Parallel VOF Method for Simulation of Water Impact on Rigid Structure”. In: *Procedia Engineering* 61 (2013), pp. 306–314.

[146] Moukalled, F. and Darwish, M. “A High-Resolution Pressure-Based Algorithm for Fluid Flow at All Speeds”. In: *Journal of Computational Physics* 168.1 (Mar. 2001), pp. 101–130.

[147] Moukalled, F., Mangani, L., and Darwish, M. “Discretization of the Convection Term”. In: *The Finite Volume Method in Computational Fluid Dynamics*. 2016. Chap. 12, p. 791.

[148] Münsch, M. “Entwicklung und Anwendung eines semi-impliziten Kopplungsverfahrens zur numerischen Berechnung der Fluid-Struktur-Wechselwirkung in turbulenten Strömungen mittels Large-Eddy Simulationen”. Doctoral thesis. Friedrich-Alexander-Universität, 2015, p. 270.

[149] Muzaferija, S. and Perić, M. “A Two-Fluid Navier-Stokes Solver to Simulate Water Entry”. In: *Proceedings of 22nd symposium on naval hydrodynamics*. National Academy Press. Washington: National Academy Press., 1999, pp. 638–651.

[150] Naudascher, E. and Rockwell, D. *Flow-induced vibrations: an engineering guide*. Courier Corporation, 2012.

[151] Ng, K. et al. “A coupled Smoothed Particle Hydrodynamics-Volume Compensated Particle Method (SPH-VCPM) for Fluid Structure Interaction (FSI) modelling”. In: *Ocean Engineering* 218.January (Dec. 2020), p. 107923.

[152] Nichols, B. and Hirt, C. “Calculating three-dimensional free surface flows in the vicinity of submerged and exposed structures”. In: *Journal of Computational Physics* 12.2 (Feb. 1973), pp. 234–246.

[153] Nicolici, S. and Bilegan, R. M. “Fluid structure interaction modeling of liquid sloshing phenomena in flexible tanks”. In: *Nuclear Engineering and Design* 258 (May 2013), pp. 51–56.

[154] Osher, S. and Sethian, J. A. “Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations”. In: *Journal of Computational Physics* 79.1 (Nov. 1988), pp. 12–49.

[155] Paik, K.-J. and Carrica, P. M. “Fluid–structure interaction for an elastic structure interacting with free surface in a rolling tank”. In: *Ocean Engineering* 84 (July 2014), pp. 201–212.

[156] Pascau, A. “Cell face velocity alternatives in a structured colocated grid for the unsteady Navier-Stokes equations”. In: *International Journal for Numerical Methods in Fluids* 65.7 (Mar. 2011), pp. 812–833.

[157] Patankar, N. et al. “A new formulation of the distributed Lagrange multiplier/fictitious domain method for particulate flows”. In: *International Journal of Multiphase Flow* 26.9 (Sept. 2000), pp. 1509–1524.

[158] Patankar, S. V. *Numerical Heat Transfer and Fluid Flow*. Taylor & Francis, Oct. 1980.

[159] Patel, J. K. and Natarajan, G. “A generic framework for design of interface capturing schemes for multi-fluid flows”. In: *Computers & Fluids* 106 (Jan. 2015), pp. 108–118.

[160] Pathak, A. and Raessi, M. “A 3D, fully Eulerian, VOF-based solver to study the interaction between two fluids and moving rigid bodies using the fictitious domain method”. In: *Journal of Computational Physics* 311 (Apr. 2016), pp. 87–113.

[161] Pedersen, P. T. “Marine Structures: Future Trends and the Role of Universities”. In: *Engineering* 1.1 (Mar. 2015), pp. 131–138.

[162] Peric, M. “A finite volume method for the prediction of three-dimensional fluid flow in complex ducts”. Ph.D. thesys. University of London, 1985.

[163] Peskin, C. S. “Numerical analysis of blood flow in the heart”. In: *Journal of Computational Physics* 25.3 (Nov. 1977), pp. 220–252.

[164] Peskin, C. S. “The immersed boundary method”. In: *Acta Numerica* 11.2002 (2002), pp. 479–517.

[165] Pin, F. D. et al. “The ALE/Lagrangian Particle Finite Element Method: A new approach to computation of free-surface flows and fluid–object interactions”. In: *Computers & Fluids* 36.1 (Jan. 2007), pp. 27–38.

[166] Prasad, B., Hino, T., and Suzuki, K. “Numerical simulation of free surface flows around shallowly submerged hydrofoil by OpenFOAM”. In: *Ocean Engineering* 102 (July 2015), pp. 87–94.

[167] Queutey, P. and Visonneau, M. “An interface capturing method for free-surface hydrodynamic flows”. In: *Computers & Fluids* 36.9 (Nov. 2007), pp. 1481–1510.

[168] Rafiee, A. and Thiagarajan, K. P. “An SPH projection method for simulating fluid-hypoelastic structure interaction”. In: *Computer Methods in Applied Mechanics and Engineering* 198.33-36 (July 2009), pp. 2785–2795.

[169] Rannacher, R. “Incompressible viscous flows”. In: *Encyclopedia of computational mechanics* (2004).

[170] Reimann, T. “Numerische Simulation von Fluid-Struktur-Interaktion in turbulenten Strömungen.” PhD thesis. TU Darmstadt, 2013.

[171] Rhie, C. M. and Chow, W. L. “Numerical study of the turbulent flow past an airfoil with trailing edge separation”. In: *AIAA Journal* 21.11 (Nov. 1983), pp. 1525–1532.

[172] Rhie, C. “A three-dimensional passage flow analysis method aimed at centrifugal impellers”. In: *Computers & Fluids* 13.4 (Jan. 1985), pp. 443–460.

[173] Rider, W. J. and Kothe, D. B. “Reconstructing Volume Tracking”. In: *Journal of Computational Physics* 141.2 (Apr. 1998), pp. 112–152.

[174] Roache, P. J. “Code Verification by the Method of Manufactured Solutions”. In: *Journal of Fluids Engineering* 124.1 (Mar. 2002), pp. 4–10.

[175] Rodriguez, S. “LES and DNS Turbulence Modeling”. In: *Applied Computational Fluid Dynamics and Turbulence Modeling*. Cham: Springer International Publishing, 2019, pp. 197–223.

[176] Roe, P. L. “Characteristic-based schemes for the Euler equations”. In: *Annual review of fluid mechanics* 18.1 (1986), pp. 337–365.

[177] Rubin, S. and Khosla, P. “Polynomial interpolation methods for viscous flow calculations”. In: *Journal of Computational Physics* 24.3 (July 1977), pp. 217–244.

[178] Rudman, M. “A volume-tracking method for incompressible multifluid flows with large density variations”. In: *International Journal for Numerical Methods in Fluids* 28.2 (1998), pp. 357–378.

[179] Rudman, M. “Volume-tracking methods for interfacial flow calculations”. In: *International Journal for Numerical Methods in Fluids* 24.7 (Apr. 1997), pp. 671–691.

[180] Sachs, S. “Multigrid Methods applied to Fluid-Structure Interaction”. PhD thesis. TU-Darmstadt, 2012.

[181] Salari, K. and Knupp, P. *Code Verification by the Method of Manufactured Solutions*. Tech. rep. Albuquerque: Sandia National Laboratories, 2000.

[182] Sauer, J. “Instationär kavitierende Strömungen: ein neues Modell, basierend auf front capturing (VoF) und Blasendynamik”. Doctoral dissertation. Universität Karlsruhe, 2000, p. 162.

[183] SCAI. *MpCCI - Mesh-Based Parallel Code Coupling Interface*. 2004.

[184] Schäfer, M. et al. “Efficient Numerical Simulation and Optimization of Fluid-Structure Interaction”. In: *Fluid Structure Interaction II*. Springer, 2011, pp. 131–158.

[185] Schäfer, M. *Computational engineering - Introduction to numerical methods*. 2006, pp. 1–321.

[186] Schäfer, M., Heck, M., and Yigit, S. “An Implicit Partitioned Method for the Numerical Simulation of Fluid-Structure Interaction”. In: *Fluid-Structure Interaction*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 171–194.

[187] Schäfer, M. and Teschauer, I. “Numerical simulation of coupled fluid–solid problems”. In: *Computer Methods in Applied Mechanics and Engineering* 190.28 (Mar. 2001), pp. 3645–3667.

[188] Sharma, A. “Level set method for computational multi-fluid dynamics: A review on developments, applications and analysis”. In: *Sadhana - Academy Proceedings in Engineering Sciences* 40.3 (2015), pp. 627–652.

[189] Staab, D. H. “Numerical treatment of multiphase flows coupled with acoustics for surface tension dominated flows”. Doctoral. Technischen Universit¨ at Darmstadt zur, 2016.

[190] Stein, K. et al. “Parachute fluid–structure interactions: 3-D computation”. In: *Computer Methods in Applied Mechanics and Engineering* 190.3-4 (Oct. 2000), pp. 373–386.

[191] Sternel, D. C. et al. “Efficiency and accuracy of fluid-structure interaction simulations using an implicit partitioned approach”. In: *Computational Mechanics* 43.1 (Dec. 2008), pp. 103–113.

[192] Stone, H. S. “An efficient parallel algorithm for the solution of a tridiagonal linear system of equations”. In: *Journal of the ACM (JACM)* 20.1 (1973), pp. 27–38.

[193] Sun, P., Le Touzé, D., and Zhang, A.-M. “Study of a complex fluid-structure dam-breaking benchmark problem using a multi-phase SPH method with APR”. In: *Engineering Analysis with Boundary Elements* 104 (July 2019), pp. 240–258.

[194] Sun, Y., Xi, G., and Sun, Z. “A fully Lagrangian method for fluid–structure interaction problems with deformable floating structure”. In: *Journal of Fluids and Structures* 90 (Oct. 2019), pp. 379–395.

[195] Sun, Z. et al. “Coupled MPS-modal superposition method for 2D nonlinear fluid-structure interaction problems with free surface”. In: *Journal of Fluids and Structures* 61 (Feb. 2016), pp. 295–323.

[196] Sun, Z., Huang, Z., and Zhou, X. “Benchmarking the material point method for interaction problems between the free surface flow and elastic structure”. In: *Progress in Computational Fluid Dynamics, An International Journal* 19.1 (2019), p. 1.

[197] Sussman, M. and Puckett, E. G. “A Coupled Level Set and Volume-of-Fluid Method for Computing 3D and Axisymmetric Incompressible Two-Phase Flows”. In: *Journal of Computational Physics* 162.2 (Aug. 2000), pp. 301–337.

[198] Sussman, M., Smereka, P., and Osher, S. “A Level Set Approach for Computing Solutions to Incompressible Two-Phase Flow”. In: *Journal of Computational Physics* 114.1 (Sept. 1994), pp. 146–159.

[199] Takizawa, K. and Tezduyar, T. E. “Computational Methods for Parachute Fluid–Structure Interactions”. In: *Archives of Computational Methods in Engineering* 19.1 (Mar. 2012), pp. 125–169.

[200] Taylor, R. L. *FEAP-A finite element analysis program*. 2000.

[201] Thomas, P. D. and Lombard, C. K. “Geometric Conservation Law and Its Application to Flow Computations on Moving Grids”. In: *AIAA Journal* 17.10 (Oct. 1979), pp. 1030–1037.

[202] Trivedi, C. “A review on fluid structure interaction in hydraulic turbines: A focus on hydrodynamic damping”. In: *Engineering Failure Analysis* 77 (July 2017), pp. 1–22.

[203] Tryggvason, G., Scardovelli, R., and Zaleski, S. *Direct Numerical Simulations of Gas–Liquid Multiphase Flows*. Cambridge: Cambridge University Press, Jan. 2001.

[204] Tsui, Y.-Y. et al. “Flux-blending schemes for interface capture in two-fluid flows”. In: *International Journal of Heat and Mass Transfer* 52.23-24 (Nov. 2009), pp. 5547–5556.

[205] Tuković, Ž. and Jasak, H. “A moving mesh finite volume interface tracking method for surface tension dominated interfacial fluid flow”. In: *Computers & Fluids* 55 (Feb. 2012), pp. 70–84.

[206] Turek, S. et al. “Numerical Benchmarking of Fluid-Structure Interaction: A Comparison of Different Discretization and Solution Approaches”. In: *Lecture Notes in Computational Science and Engineering*. 2011, pp. 413–424. arXiv: arXiv:1011.1669v3.

[207] Turek, S. and Hron, J. “Proposal for numerical benchmarking of fluid-structure interaction between an elastic object and laminar incompressible flow”. In: *Fluid-structure interaction*. Dortmund: Springer Berlin Heidelberg, 2006, pp. 371–385.

[208] Turek, S., Mierka, O., and Bäumler, K. “Numerical Benchmarking for 3D Multiphase Flow: New Results for a Rising Bubble”. In: *Numerical Mathematics and Advanced Applications ENUMATH 2017*. 2019, pp. 593–601.

[209] Ubbink, O. and Issa, R. “A Method for Capturing Sharp Fluid Interfaces on Arbitrary Meshes”. In: *Journal of Computational Physics* 153.1 (July 1999), pp. 26–50.

[210] Ubbink, O. “Numerical prediction of two fluid systems with sharp interfaces”. Doctoral. Universtity of London, 1997.

[211] Udaykumar, H. et al. “Multiphase Dynamics in Arbitrary Geometries on Fixed Cartesian Grids”. In: *Journal of Computational Physics* 137.2 (Nov. 1997), pp. 366–405.

[212] Uekermann, B. “Partitioned Fluid-Structure Interaction on Massively Parallel Systems”. Doctoral dissertation. Technischen Universit¨at M¨unchen, 2016.

[213] Uekermann, B. et al. “Official preCICE Adapters for Standard Open-Source Solvers”. In: *Proceedings of the 7th GACM Colloquium on Computational Mechanics for Young Scientists from Academia* (2017).

[214] Unverdi, S. O. and Tryggvason, G. “A front-tracking method for viscous, incompressible, multi-fluid flows”. In: *Journal of Computational Physics* 100.1 (May 1992), pp. 25–37.

[215] Van Der Pijl, S. P. et al. “A mass-conserving Level-Set method for modelling of multi-phase flows”. In: *International Journal for Numerical Methods in Fluids* 47.4 (Feb. 2005), pp. 339–361.

[216] Van Doormaal, J. P. and Raithby, G. D. “Enhancements of the SIMPLE method for predicting incompressible fluid flows”. In: *Numerical Heat Transfer* 7.2 (Apr. 1984), pp. 147–163.

[217] Van Leer, B. “Towards the ultimate conservative difference scheme. V. A second-order sequel to Godunov’s method”. In: *Journal of Computational Physics* 32.1 (July 1979), pp. 101–136.

[218] Van Loon, R., Anderson, P., and Van de Vosse, F. “A fluid–structure interaction method with solid-rigid contact for heart valve dynamics”. In: *Journal of Computational Physics* 217.2 (Sept. 2006), pp. 806–823.

[219] Van Loon, R. et al. “A combined fictitious domain/adaptive meshing method for fluid–structure interaction in heart valves”. In: *International Journal for Numerical Methods in Fluids* 46.5 (Oct. 2004), pp. 533–544.

[220] Van Zuijlen, A., Bosscher, S., and Bijl, H. “Two level algorithms for partitioned fluid–structure interaction computations”. In: *Computer Methods in Applied Mechanics and Engineering* 196.8 (Jan. 2007), pp. 1458–1470.

[221] Vedovoto, J. M. et al. “Application of the method of manufactured solutions to the verification of a pressure-based finite-volume numerical scheme”. In: *Computers & Fluids* 51.1 (Dec. 2011), pp. 85–99.

[222] Vierendeels, J. et al. “Implicit coupling of partitioned fluid–structure interaction problems with reduced order models”. In: *Computers & Structures* 85.11-14 (June 2007), pp. 970–976.

[223] Viré, A., Xiang, J., and Pain, C. C. “An immersed-shell method for modelling fluid–structure interactions”. In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 373.2035 (Feb. 2015), p. 20140085.

[224] Wackers, J. et al. “Free-Surface Viscous Flow Solution Methods for Ship Hydrodynamics”. In: *Archives of Computational Methods in Engineering* 18.1 (Mar. 2011), pp. 1–41.

[225] Wacławczyk, T., Caner Gemici, Ö., and Schäfer, M. “Novel high-resolution scheme for interface capturing in multi-phase flow”. In: *6th International Conference on Multiphase Flow*. Leipzig, 2007.

[226] Walhorn, E. et al. “Fluid–structure coupling within a monolithic model involving free surface flows”. In: *Computers & Structures* 83.25-26 (Sept. 2005), pp. 2100–2111.

[227] Wall, W. A. et al. “Partitioned Analysis of Transient Nonlinear Fluid Structure Interaction Problems Including Free Surface Effects”. In: *Multifield Problems: State of the Art*. Ed. by A.-M. Sändig, W. Schiehlen, and W. L. Wendland. Berlin, Heidelberg: Springer Berlin Heidelberg, 2000, pp. 159–166.

[228] Wall, W. A., Genkinger, S., and Ramm, E. “A strong coupling partitioned approach for fluid–structure interaction with free surfaces”. In: *Computers & Fluids* 36.1 (Jan. 2007), pp. 169–183.

[229] Wei, G. and Kirby, J. T. “Time-Dependent Numerical Code for Extended Boussinesq Equations”. In: *Journal of Waterway, Port, Coastal, and Ocean Engineering* 121.5 (Sept. 1995), pp. 251–261.

[230] Wesseling, P. *Principles of Computational Fluid Dynamics*. Vol. 29. Springer Science & Business Media, 2009.

[231] Wu, T.-R. et al. “A two-way coupled simulation of moving solids in free-surface flows”. In: *Computers & Fluids* 100 (Sept. 2014), pp. 347–355.

[232] Xiao, F., Ii, S., and Chen, C. “Revisit to the THINC scheme: A simple algebraic VOF algorithm”. In: *Journal of Computational Physics* 230.19 (Aug. 2011), pp. 7086–7092.

[233] Xiao, H. et al. “Numerical modeling of wave–current forces acting on horizontal cylinder of marine structures by VOF method”. In: *Ocean Engineering* 67 (July 2013), pp. 58–67.

[234] Yang, C. and Zhang, H. “Numerical simulation of the interactions between fluid and structure in application of the MPS method assisted with the large eddy simulation method”. In: *Ocean Engineering* 155.November 2016 (May 2018), pp. 55–64.

[235] Yang, L. “One-fluid formulation for fluid–structure interaction with free surface”. In: *Computer Methods in Applied Mechanics and Engineering* 332 (Apr. 2018), pp. 102–135.

[236] Yang, P. et al. “Modelling of fluid–structure interaction with multiphase viscous flows using an immersed-body method”. In: *Journal of Computational Physics* 321 (Sept. 2016), pp. 571–592.

[237] Yang, Q., Jones, V., and McCue, L. “Free-surface flow interactions with deformable structures using an SPH–FEM model”. In: *Ocean Engineering* 55 (Dec. 2012), pp. 136–147.

[238] Yapici, K. and Uludag, Y. “Finite volume simulation of 2-D steady square lid driven cavity flow at high reynolds numbers”. In: *Brazilian Journal of Chemical Engineering* 30.4 (Dec. 2013), pp. 923–937.

[239] Yen, R.-H. and Liu, C.-H. “Enhancement of the SIMPLE algorithm by an additional explicit corrector step”. In: *Numerical Heat Transfer, Part B: Fundamentals* 24.1 (July 1993), pp. 127–141.

[240] Yigit, S. et al. “An enhanced implicit partitioned ALE aproach for fluid-structure interaction simulations”. In: *Computational methods for coupled problems in science and engineering*. Barcelona, 2007.

[241] Yigit, S., Schäfer, M., and Heck, M. “Grid movement techniques and their influence on laminar fluid–structure interaction computations”. In: *Journal of Fluids and Structures* 24.6 (Aug. 2008), pp. 819–832.

[242] You, S. and Bathe, K.-J. “Transient solution of 3D free surface flows using large time steps”. In: *Computers & Structures* 158 (Oct. 2015), pp. 346–354.

[243] Yu, B. et al. “Discussion on momentum interpolation method for collocated grids of incompressible flow”. In: *Numerical Heat Transfer, Part B: Fundamentals* 42.2 (Aug. 2002), pp. 141–166.

[244] Zalesak, S. T. “Fully multidimensional flux-corrected transport algorithms for fluids”. In: *Journal of Computational Physics* 31.3 (June 1979), pp. 335–362.

[245] Zaleski, S. “Simulations of high Reynolds number breakup of liquid-gas interfaces”. In: *Lecture series-van Kareman Institute for fluid dynamics*. 1996. Chap. 2, pp. L1–L31.

[246] Zhang, C. et al. “A sharp interface immersed boundary/VOF model coupled with wave generating and absorbing options for wave-structure interaction”. In: *Computers & Fluids* 89 (Jan. 2014), pp. 214–231.

[247] Zhang, G., Chen, X., and Wan, D. “MPS-FEM Coupled Method for Study of Wave-Structure Interaction”. In: *Journal of Marine Science and Application* 18.4 (2019), pp. 387–399.

[248] Zhang, G. et al. “Coupling of SPH with smoothed point interpolation method for violent fluid-structure interaction problems”. In: *Engineering Analysis with Boundary Elements* 103.February (June 2019), pp. 1–10.

[249] Zhang, Y. and Wan, D. “MPS-FEM Coupled Method for Fluid-Structure Interaction in 3D Dam-Break Flows”. In: *International Journal of Computational Methods* 16.2 (2019).

[250] Zhao, L. et al. “Improved conservative level set method”. In: *International Journal for Numerical Methods in Fluids* 75.8 (July 2014), pp. 575–590.

[251] Zhu, L. and Peskin, C. S. “Simulation of a Flapping Flexible Filament in a Flowing Soap Film by the Immersed Boundary Method”. In: *Journal of Computational Physics* 179.2 (July 2002), pp. 452–468.

[252] Zhu, Q. et al. “An immersogeometric formulation for free-surface flows with application to marine engineering problems”. In: *Computer Methods in Applied Mechanics and Engineering* 361 (Apr. 2020), p. 112748.

A. Appendix

A.1. The DABT scheme

The DABT scheme uses ten central nodes in the approximation of face derivatives and does not incorporate any interpolation. For instance, the DABT expression for the derivative of a variable ϕ at face e based on the geometric configuration shown in Figure A.1 has the form

$$\left(\frac{\partial \phi}{\partial x_i}\right)_e \approx \frac{\beta_{ij}^{e,DABT}}{J^{e,DABT}} \left(\Delta\phi\right)_{e,DABT}^{j}, \quad i,j = 1,2,3, \tag{A.1}$$

where the Jacobian determinant $J^{e,DABT}$ and the adjoint matrix $\beta_{ij}^{e,DABT}$ belong to the transformation matrix $\mathbf{B}_{ij}^{e,DABT}$ given by

$$\mathbf{B}_{ij}^{e,DABT} = \begin{bmatrix} (x_E - x_P) & (x_N - x_S + x_{NE} - x_{SE}) & (x_T - x_B + x_{TE} - x_{BE}) \\ (y_E - y_P) & (y_N - y_S + y_{NE} - y_{SE}) & (y_T - y_B + y_{TE} - y_{BE}) \\ (z_E - z_P) & (z_N - z_S + z_{NE} - z_{SE}) & (z_T - z_B + z_{TE} - z_{BE}) \end{bmatrix}, \tag{A.2}$$

and the variation of the variable ϕ is defined as

$$\left(\Delta\phi\right)_{e,DABT}^{j} = \begin{bmatrix} (\phi_E - \phi_P) & (\phi_N - \phi_S + \phi_{NE} - \phi_{SE}) & (\phi_T - \phi_B + \phi_{TE} - \phi_{BE}) \end{bmatrix}. \tag{A.3}$$

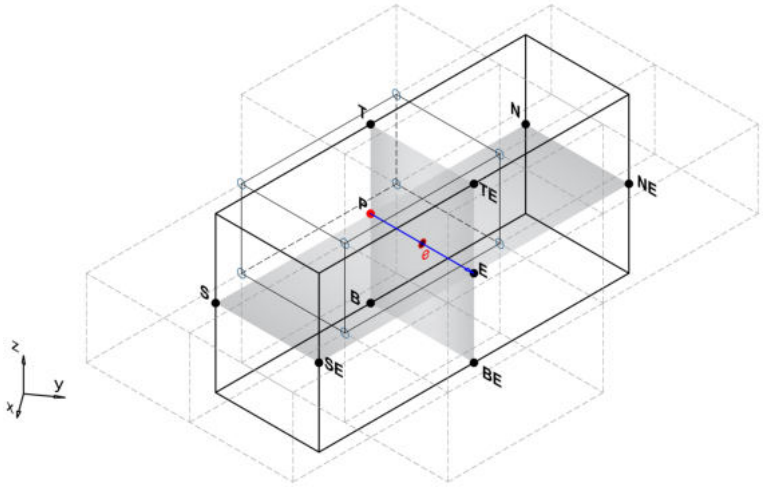


Figure A.1.: The auxiliary control volume around the east face e used by the DABT scheme to approximate the face derivate

B. Appendix

B.1. The Green Gauss gradient with the midpoint correction for non-orthogonal grids

The method has been extracted from reference [147]

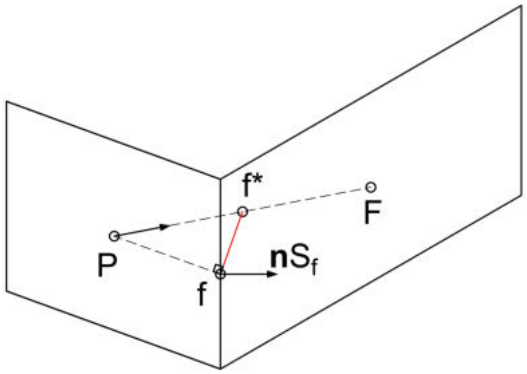


Figure B.1.: The connection between two non-orthogonal control volumes P and F using the midpoint approach in a two-dimensional configuration

The gradient field calculation of a variable φ over the domain is an iterative process that proceeds as below:

During the first iteration:

1. Approximate the face values as
$$\varphi_{f*} = \frac{\varphi_P + \varphi_F}{2}. \tag{B.1}$$
2. Compute the nodal gradient using the Green Gauss equation
$$\nabla\varphi_P = \frac{1}{\delta V_P} \sum_{f(P)} \varphi_{f*} \mathbf{n} S_f. \tag{B.2}$$

From the second iteration onward, correct the gradient field with the following extra steps:

3. Update the face values with
$$\varphi_f = \varphi_{f*} + \frac{[\nabla\varphi_P + \nabla\varphi_F]}{2} \cdot \left[\mathbf{x}_f - \frac{\mathbf{x}_P + \mathbf{x}_F}{2}\right], \tag{B.3}$$

where $\mathbf{x}_f$, $\mathbf{x}_P$, and $\mathbf{x}_F$ represent the vector position of the face center, centroid of cell P, and centroid of cell F, respectively.

4. Update the nodal gradient using the new face values

$$\nabla\varphi_P = \frac{1}{\delta V_P}\sum_{f(P)} \varphi_f \mathbf{n} S_f \tag{B.4}$$

5. Go back to step **3** and repeat until achieving a consistent value for $\nabla\varphi_P$.

In the FASTEST code, the iteration process is set up to make two iterations.

C. Appendix

C.1. Linearization factors ℓ and m for some interface capturing schemes formulated in the NVF framework. All apply for uniform grids

Table C.1.: Linearization factors ℓ and m.

Begin of Table

NVF form	Linearization factors
CICSAM[209]	
Compressive scheme: HYPER-C (CBC)	
$\tilde{\alpha}_f^{\text{CBC}} = \begin{cases} \min\left(\frac{\tilde{\alpha}_D}{Co_D}, 1\right) & 0 \le \tilde{\alpha}_D \le 1 \\ \tilde{\alpha}_D & \text{otherwise} \end{cases}$	$[\ell, m]^{\text{CBC}} = \begin{cases} \left[\frac{1}{Co_D}, 0\right] & 0 \le \tilde{\alpha}_D \le 1, \quad \frac{\tilde{\alpha}_D}{Co_D} < 1 \\ [0, 1] & 0 \le \tilde{\alpha}_D \le 1, \quad \frac{\tilde{\alpha}_D}{Co_D} \ge 1 \\ [1, 0] & \text{otherwise} \end{cases}$
Diffusive HR scheme: ULTIMATE-QUICKEST (UQ)	
$\tilde{\alpha}_f^{\text{UQ}} = \begin{cases} \min\left[\frac{1}{4}\left(Co_D + 3\right)\tilde{\alpha}_D + \frac{3}{8}\left(1 - Co_D\right), \tilde{\alpha}_f^{\text{CBC}}\right] & 0 \le \tilde{\alpha}_D \le 1 \\ \tilde{\alpha}_D & \text{otherwise} \end{cases}$	$[\ell, m]^{\text{UQ}} = \begin{cases} \min\left\{\left[\frac{1}{4}\left(Co_D + 3\right), \frac{3}{8}\left(1 - Co_D\right)\right], [\ell, m]^{\text{CBC}}\right\} & 0 \le \tilde{\alpha}_D \le 1 \\ [1, 0] & \text{otherwise} \end{cases}$
$\tilde{\alpha}_f^{\text{CICSAM}} = \lambda \tilde{\alpha}_f^{\text{CBC}} + (1 - \lambda)\,\tilde{\alpha}_f^{\text{UQ}}$ with $\lambda_f = \min\left\{\frac{\cos(2\theta_f)+1}{2}, 1\right\}$	$[\ell, m]^{\text{CIMSAM}} = \lambda_f [\ell, m]^{\text{CBC}} + (1 - \lambda_f)\,[\ell, m]^{\text{UQ}}$
HRIC[149]	
Compressive scheme: Bounded downwind (BD)	
$\tilde{\alpha}_f^{\text{BD}} = \begin{cases} 2\tilde{\alpha}_D & 0 \le \tilde{\alpha}_D < \frac{1}{2} \\ 1 & \frac{1}{2} \le \tilde{\alpha}_D \le 1 \\ \tilde{\alpha}_D & \text{otherwise} \end{cases}$	$[\ell, m]^{\text{BD}} = \begin{cases} [2, 0] & 0 \le \tilde{\alpha}_D \le \frac{1}{2} \\ [0, 1] & \frac{1}{2} \le \tilde{\alpha}_D \le 1 \\ [1, 0] & \text{otherwise} \end{cases}$
Diffusive scheme: Upwind differencing (DU)	
$\tilde{\alpha}_f^{\text{UD}} = \tilde{\alpha}_D$	$[\ell, m]^{\text{UD}} = [1, 0]$
$\tilde{\alpha}_f^{*} = \lambda \tilde{\alpha}_f^{\text{BD}} + (1 - \lambda)\,\tilde{\alpha}_f^{\text{UD}}$ with $\lambda_f = \lvert\cos\theta_f\rvert^{1/2}$	$[\ell, m]^{*} = \lambda_f [\ell, m]^{\text{BD}} + (1 - \lambda_f)\,[\ell, m]^{\text{UD}}$
Extra correction due to Co value	

Continuation of Table C.1

NVF form	Linearization factors
MCICSAM-W[Wacawczyk 2007]	
Compressive scheme: the steady version of HYPER-C (CBC)	
$\tilde{\alpha}_f^{\text{CBC}} = \begin{cases} \min(2\tilde{\alpha}_D, 1) & 0 \le \tilde{\alpha}_D \le 1 \\ \tilde{\alpha}_D & \text{otherwise} \end{cases}$	$[\ell, m]^{\text{CBC}} = \begin{cases} [2,0] & 0 \le \tilde{\alpha}_D \le 1, \quad 2\tilde{\alpha}_D < 1 \\ [0,1] & 0 \le \tilde{\alpha}_D \le 1, \quad 2\tilde{\alpha}_D \ge 1 \\ [1,0] & \text{otherwise} \end{cases}$
Diffusive HR scheme: FROMM (FR)	
$\tilde{\alpha}_f^{\text{FR}} = \begin{cases} \min\left(\frac{1}{4} + \tilde{\alpha}_D, \tilde{\alpha}_f^{CBC}\right) & 0 \le \tilde{\alpha}_D \le 1 \\ \tilde{\alpha}_D & \text{otherwise} \end{cases}$	$[\ell, m]^{\text{FR}} = \begin{cases} \min\left\{\left[1, \frac{1}{4}\right], [\ell, m]^{\text{CBC}}\right\} & 0 \le \tilde{\alpha}_D \le 1 \\ [1,0] & \text{otherwise} \end{cases}$
$\tilde{\alpha}_f^{\text{CICSAM-W}} = \lambda \tilde{\alpha}_f^{\text{CBC}} + (1-\lambda)\,\tilde{\alpha}_f^{\text{FR}}$ with $\lambda_f = \lvert\cos(2\theta_f)\rvert^{\frac{1}{4}}$	$[\ell, m]^{\text{CICSAM-W}} = \lambda_f [\ell, m]^{\text{CBC}} + (1-\lambda_f)\,[\ell, m]^{\text{FR}}$
MCICSAM-Z[246]	
Compressive scheme: Compressive differencing scheme (C) based on SUPERBEE (S)	
$\alpha_f^{\text{C}} = \begin{cases} \min\left(\frac{\tilde{\alpha}_D}{Co_D}, 1\right) & 0 \le \tilde{\alpha}_D \le 1, \quad 0 < Co_D \le 0.3 \\ \min\left(\frac{\tilde{\alpha}_D}{0.3}, 1\right) & 0 \le \tilde{\alpha}_D \le 1, \quad 0.3 < Co_D \le 0.6 \\ a \min\left(\frac{\tilde{\alpha}_D}{0.3}, 1\right) + b\tilde{\alpha}_f^S & 0 \le \tilde{\alpha}_D \le 1, \quad 0.6 < Co_D \le 0.7 \\ \tilde{\alpha}_f^S & 0 \le \tilde{\alpha}_D \le 1, \quad Co_D > 0.7 \\ \tilde{\alpha}_D & \text{otherwise} \end{cases}$	$[\ell, m]^{\text{C}} = \begin{cases} \left[\frac{1}{Co_D}, 0\right] & 0 \le \tilde{\alpha}_D \le 1, \quad \frac{\tilde{\alpha}_D}{Co_D} < 1, \quad 0 < Co_D \le 0.3 \\ [0,1] & 0 \le \tilde{\alpha}_D \le 1, \quad \frac{\tilde{\alpha}_D}{Co_D} \ge 1, \quad 0 < Co_D \le 0.3 \\ \left[\frac{1}{0.3}, 0\right] & 0 \le \tilde{\alpha}_D \le 1, \quad \frac{\tilde{\alpha}_D}{0.3} < 1, \quad 0.3 < Co_D \le 0.6 \\ [0,1] & 0 \le \tilde{\alpha}_D \le 1, \quad \frac{\tilde{\alpha}_D}{0.3} \ge 1, \quad 0.3 < Co_D \le 0.6 \\ a\left[\frac{1}{0.3}, 0\right] + b[\ell, m]^{\text{S}} & 0 \le \tilde{\alpha}_D \le 1, \quad \frac{\tilde{\alpha}_D}{0.3} < 1, \quad 0.6 < Co_D \le 0.7 \\ a[0,1] + b[\ell, m]^{\text{S}} & 0 \le \tilde{\alpha}_D \le 1, \quad \frac{\tilde{\alpha}_D}{0.3} \ge 1, \quad 0.6 < Co_D \le 0.7 \\ [\ell, m]^{\text{S}} & 0 \le \tilde{\alpha}_D \le 1, \quad Co_D > 0.7 \\ [1,0] & \text{otherwise} \end{cases}$
with $a = \frac{0.7 - Co_D}{0.1}$, $b = \frac{Co_D - 0.6}{0.1}$	
$\tilde{\alpha}_f^{\text{S}} = \begin{cases} 2\tilde{\alpha}_D & 0 \le \tilde{\alpha}_D < \frac{1}{3} \\ \frac{1}{2}\tilde{\alpha}_D + \frac{1}{2} & \frac{1}{3} \le \tilde{\alpha}_D < \frac{1}{2} \\ \frac{3}{2}\tilde{\alpha}_D & \frac{1}{2} \le \tilde{\alpha}_D < \frac{2}{3} \\ 1 & \frac{2}{3} \le \tilde{\alpha}_D \le 1 \\ \tilde{\alpha}_D & \text{otherwise} \end{cases}$	$[\ell, m]^{\text{S}} = \begin{cases} [2,0] & 0 \le \tilde{\alpha}_D < \frac{1}{3} \\ \left[\frac{1}{2}, \frac{1}{2}\right] & \frac{1}{3} \le \tilde{\alpha}_D < \frac{1}{2} \\ \left[\frac{3}{2}, 0\right] & \frac{1}{2} \le \tilde{\alpha}_D < \frac{2}{3} \\ [0,1] & \frac{2}{3} \le \tilde{\alpha}_D \le 1 \\ [1,0] & \text{otherwise} \end{cases}$
Diffusive HR scheme: MUSCL	
$\tilde{\alpha}_f^{\text{MUSCL}} = \begin{cases} 2\tilde{\alpha}_D & 0 \le \tilde{\alpha}_D < \frac{1}{4} \\ \tilde{\alpha}_D + \frac{1}{4} & \frac{1}{4} \le \tilde{\alpha}_D < \frac{3}{4} \\ 1 & \frac{3}{4} \le \tilde{\alpha}_D < \frac{1}{4} \\ \tilde{\alpha}_D & otherwise \end{cases}$	$[\ell, m]^{\text{MUSCL}} = \begin{cases} [2,0] & 0 \le \tilde{\alpha}_D < \frac{1}{4} \\ \left[1, \frac{1}{4}\right] & \frac{1}{4} \le \tilde{\alpha}_D < \frac{3}{4} \\ [0,1] & \frac{3}{4} \le \tilde{\alpha}_D < \frac{1}{4} \\ [1,0] & otherwise \end{cases}$
$\tilde{\alpha}_f^{\text{MCICSAM-Z}} = \lambda_{f-1} \tilde{\alpha}_f^{\text{lower-bound}} + (1-\lambda_{f-1})\,\tilde{\alpha}_f^{\text{upper-bound}}$ with	$[\ell, m]^{\text{MCICSAM-Z}} = \lambda_{f-1} [\ell, m]^{\text{lower-bound}} + (1-\lambda_{f-1})\,[\ell, m]^{\text{upper-bound}}$
$\lambda_{f-1} = \begin{cases} 1 & \text{if } \tilde{\alpha}_{f-1} \le \tilde{\alpha}_D \le \tilde{\alpha}_{f+1} \\ 0 & \text{otherwise} \end{cases}$	
$\tilde{\alpha}_f^{\text{lower-bound}} = \begin{cases} \tilde{\alpha}_{f-1} & \text{if } \tilde{\alpha}_{f-1} \le \tilde{\alpha}_{f+1} \\ \tilde{\alpha}_{f+1} & \text{otherwise} \end{cases}$	$[\ell, m]^{\text{lower-bound}} = \begin{cases} [\ell, m]^{\text{MUSCL}}_{f-1} & \text{if } \tilde{\alpha}_{f-1} \le \tilde{\alpha}_{f+1} \\ [\ell, m]^{\text{MUSCL}}_{f+1} & \text{otherwise} \end{cases}$
$\tilde{\alpha}_f^{\text{upper-bound}} = \begin{cases} \tilde{\alpha}_{f+1} & \text{if } \tilde{\alpha}_{f-1} \le \tilde{\alpha}_{f+1} \\ \tilde{\alpha}_{f-1} & \text{otherwise} \end{cases}$	$[\ell, m]^{\text{upper-bound}} = \begin{cases} [\ell, m]^{\text{MUSCL}}_{f+1} & \text{if } \tilde{\alpha}_{f-1} \le \tilde{\alpha}_{f+1} \\ [\ell, m]^{\text{MUSCL}}_{f-1} & \text{otherwise} \end{cases}$

Continuation of Table C.1

NVF form	Linearization factors
$\tilde{\alpha}_f^{\text{upper-bound}} = \lambda_{f-2-u}\tilde{\alpha}_f^{\text{C}} + (1-\lambda_{f-2-u})\,\tilde{\alpha}_f^{\text{MUSCL}}$	$[\ell,m]^{\text{upper-bound}} = \lambda_{f-2-u}[\ell,m]^{\text{C}} + (1-\lambda_{f-2-u})\,[\ell,m]^{\text{MUSCL}}$
$\tilde{\alpha}_f^{\text{lower-bound}} = \lambda_{f-2-l}\tilde{\alpha}_f^{\text{C}} + (1-\lambda_{f-2-l})\,\tilde{\alpha}_f^{\text{MUSCL}}$	$[\ell,m]^{\text{lower-bound}} = \lambda_{f-2-l}[\ell,m]^{\text{C}} + (1-\lambda_{f-2-l})\,[\ell,m]^{\text{MUSCL}}$
For more details about λ_{f-1}, λ_{f-2-u}, and λ_{f-2-l} refer to the original paper.	
FBICS[204]	
Compressive scheme: Bounded downwind (BD)	
$\tilde{\alpha}_f^{\text{BD}} = \begin{cases} 3\tilde{\alpha}_D & 0 < \tilde{\alpha}_D \le \frac{1}{3} \\ 1 & \frac{1}{3} < \tilde{\alpha}_D \le 1 \\ \tilde{\alpha}_D & \text{otherwise} \end{cases}$	$[\ell,m]^{\text{BD}} = \begin{cases} [3,0] & 0 < \tilde{\alpha}_D \le \frac{1}{3} \\ [0,1] & \frac{1}{3} < \tilde{\alpha}_D \le 1 \\ [1,0] & \text{otherwise} \end{cases}$
Diffusive HR scheme:	
$\tilde{\alpha}_f^{\text{HR}} = \begin{cases} 3\tilde{\alpha}_D & 0 < \tilde{\alpha}_D \le \frac{1}{8} \\ \tilde{\alpha}_D + \frac{1}{4} & \frac{1}{8} < \tilde{\alpha}_D \le \frac{3}{4} \\ 1 & \frac{3}{4} < \tilde{\alpha}_D \le 1 \\ \tilde{\alpha}_D & otherwise \end{cases}$	$[\ell,m]^{\text{HR}} = \begin{cases} [3,0] & 0 < \tilde{\alpha}_D \le \frac{1}{8} \\ \left[1,\frac{1}{4}\right] & \frac{1}{8} < \tilde{\alpha}_D \le \frac{3}{4} \\ [0,1] & \frac{3}{4} < \tilde{\alpha}_D \le 1 \\ [1,0] & otherwise \end{cases}$
$\tilde{\alpha}_f^{\text{FBICS}} = \lambda\tilde{\alpha}_f^{\text{BD}} + (1-\lambda)\,\tilde{\alpha}_f^{\text{HR}}$ with $\lambda_f = \lvert\cos\theta_f\rvert^4$	$[\ell,m]^{\text{FBICS}} = \lambda_f[\ell,m]^{\text{BD}} + (1-\lambda_f)\,[\ell,m]^{\text{HR}}$

End of Table

D. Appendix

D.1. preCICE configuration file for the FSI 3 test case

```
<?xml version="1.0"?>
<precice-configuration>
  <log>
    <sink filter="%Severity%>debug and %Rank%=0" format="---[precice]
%ColorizedSeverity% %Message%" enabled="true"/>
  </log>

  <solver-interface dimensions="3">
  <data:vector name="Forces"/>
  <data:vector name="Displacements"/>

  <mesh name="Fluid_Nodes">
    <use-data name="Displacements"/>
  </mesh>
  <mesh name="Fluid_Centers">
    <use-data name="Forces"/>
  </mesh>
  <mesh name="Structure_Nodes">
    <use-data name="Forces"/>
    <use-data name="Displacements"/>
  </mesh>

  <participant name="FASTEST">
    <use-mesh name="Fluid_Nodes" provide="yes"/>
    <use-mesh name="Fluid_Centers" provide="yes"/>
    <use-mesh name="Structure_Nodes" from="Calculix"/>
    <write-data name="Forces" mesh="Fluid_Centers"/>
    <read-data name="Displacements" mesh="Fluid_Nodes"/>
    <mapping:rbf-thin-plate-splines direction="write" from="Fluid_Centers"
to="Structure_Nodes"constraint="conservative" z-dead="true" />
    <mapping:rbf-thin-plate-splines direction="read"from="Structure_Nodes"
to="Fluid_Nodes" constraint="consistent"/>
  </participant>
```

```
  <participant name="Calculix">
    <use-mesh name="Structure_Nodes" provide="yes"/>
    <write-data name="Displacements" mesh="Structure_Nodes"/>
    <read-data name="Forces" mesh="Structure_Nodes"/>
    <watch-point mesh="Structure_Nodes" name="point␣A"
     coordinate="0.6;0.2;0." />
  </participant>

  <m2n:sockets from="FASTEST" to="Calculix" exchange-directory="../"/>

  <coupling-scheme:serial-implicit>
    <participants first="FASTEST" second="Calculix"/>
    <time-window-size value="2.e-3" />
    <max-time value="5"/>
    <max-iterations value="50"/>
    <exchange data="Displacements" mesh="Structure_Nodes" from="Calculix"
to="FASTEST"/>
    <exchange data="Forces" mesh="Structure_Nodes" from="FASTEST"
to="Calculix"/>
    <relative-convergence-measure limit="1e-6" data="Displacements"
mesh="Structure_Nodes"/>
    <relative-convergence-measure limit="1e-2" data="Forces"
mesh="Structure_Nodes"/>
    <extrapolation-order value="2"/>
    <acceleration:constant>
        <relaxation value="0.2"/>
    </acceleration:constant>
  </coupling-scheme:serial-implicit>

</solver-interface>
</precice-configuration>
```